DC3 Challenge 2008
SOLUTION PACKET

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

October 29, 2008

Communication on 10/20/2008 with challenge@dc3.mil verified that “Typed solutions that mimic the packet sheets will be sufficient” for submission. This report takes place of the written sheets and includes notes on methodology, answers to challenges, and tools used. There are also several appendixes that list longer outputs of tools and source code for new tools developed.

Please also find a CD-ROM titled “DC3 2008 Submission – f0gd0gs” which contains source code, examples, recovered files and other files associated with this report.

Contents

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

101 Detect Suspicious Software

Methodology:

Start with a known good virtual machine. Run COTS tools that detect “suspicious software” on the known good machine to eliminate any false positives. Insert the provided data into the virtual machine and re-run the tools. Prior to running each tool the virtual machine is reset to the original state and the provided data is re-inserted.

Simply using the vmware-mount utility to copy all provided files into the Windows\System32\config directory resulted in the virtual machine not being able to boot (with blue screens or lsass errors). However, if only the SOFTARE hive is copied, the virtual machine could boot. More work should be done to conclude the best method of “inserting” the provided registry data into a working instance of Windows. Also, since much of the state is being inserted, things like login passwords become an issue (for example the provided registry files have a TallTower account that requires a password). If SAM was not replaced, the password information for stored accounts is not available.

The registry files were copied to the VM using vmware’s vmware-mount utiliy. Then the provided hives were imported using Windows’ built in regedt32 utility. Then a snapshot of the VM was taken to facilitate faster testing. The adware and spyware tools listed below were then each installed and run, restoring the VM between each trial. The tools were each run in “full” mode (or equivalent).

Either approach will likely be most successfully when using a VM that has the same Operating System that the provided files came from originally. Since the importing of registry items will result in the location of the item not being in their original place and since the tools are largely signature based it is expected that the first approach would have higher accuracy.

Answers:

Due to time issues, the above approaches were never fully realized. By manual inspection the following keys were observed:

SOFTWARE\EvidenceEliminator5.0

This is a well known program that attempts to cleans a system of potential “evidence”

Run, runonce, internet explorer and other popular keys that are installed / used by spyware appear to be “normal.”

Tools used:
Type: Spyware removal
Name: Adaware Free
Publisher: lavasoft
Commercial
Site: www.lavasoft.com

Type: Spyware removal
Name: Hijack this
Publisher: trend micro
Commercial / free
Site: http://us.trendmicro.com/us/products/personal/free-tools-and-services.html

Type: VMware utility
Name: vmware-mount
Publisher: VMWare
Commercial / free
Site: www.vmware.com/download/eula/diskmount_ws_v55.html

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

102 Hash Analysis

Methodology:

Use hashing tools commonly found on Open Source Operating systems such as Linux to hash and sort all the provided files. The following two commands can be used to quickly assess all the files using md5 and sha1 algorithms:

```
#!/bin/bash
echo "listing all non-unique md5sums"
md5sum ./*/* | sort | uniq -D -w 32
echo "listing all non-unique sha1sums"
sha1sum ./*/* | sort | uniq -D -w 40
```

Similarly you could use the provided tool like this (match_filter must be compiled):

```
md5sum ./*/* | sort | ./match_filter
sha1sum ./*/* | sort | ./match_filter
```

If you want to use a dedicated forensics product, you could used EnCase (tested on Version 5) to create a Hash Set by selected all files in the "files_from_hard_drive" directory, right-clicking and choose "Create Hash Set". It is recommended that you choose "Notable" as the category). Go to the Hash Sets pane: click view->Hash Sets. Select the newly created Hash Set, right click and "rebuild the library." In the Entries pane, select only the three USB directories. Execute a Search with "Compute Hash Values" and "recomputed Hash Values" selected. All the matching files are indicated in the Hash Set column, though no information about which files are duplicates is presented. Practically, you could also just "home plate" the entire 102 directory then sort the view my hash value. This allows you to observe which USB file is identical to which hard disk file.

Answers:

The hash analysis indicates that:

a1, and a2 on the HD are 238 from USB2
f1, and f2 on the HD are 100 from USB2
h1 and h2 on the HD are 112 and 279 from USB2

even though there is no indication that they came from a USB drive:
k1 and k3 are the same
b1 and b3 are the same
c1 and c2 are the same
d2 and d3 are the same
e1 and e2 are the same
g1 and g3 are the same
j1 and j3 are the same

Tools used:	md5sum and sha1sum build into Fedora Core 8
Type:	Basic Utility
Name:	md5sum / sha1sum
Publisher:	Fedora Community / FSF
License:	Open Source (GPL)
Site:	http://www.gnu.org/software/coreutils/

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

103 – Image Analysis

Methodology:

This task is very broad. Technically all the pictures contain meta-data. Most (all?) of the information present in the file headers can be considered meta data. The “of investigative interest” column is debatable and in reality would heavily depend upon the particular case. In one case answering a question like “did the flash fire” or “what was the focal length of the lens” or “what were the GPS coordinates” might be very important, in others that information might not be pertinent at all.

Common items of investigative interest involve filename, filetype, dates, times, camera make/model, Software, User comment, Photoshop, Adobe, Acquire, Creator, owner, Profile, Derived, Device, Firmware, uuid, filenumber, serial, and camera. Also popular brands of cameras may be of interest: Casio, Nikon, sony, canon, Olympus, Minolta, pentax, Kodak, leica, HP, Samsung, sanyo, etc.

I good first cut would be to run:

```
#> for i in `ls *`; do exiftool –v $i > $i.txt; done;
```

(your could try exiftool –U –v5, but the volume of output is much greater)

Then filter through the files looking for case specific items of interest.

An easy way to filter the exiftool output based keywords would be:

```
#> egrep –nihf interesting.dat *.txt
```

(assuming you’ve already put exif output into txt files, and “interesting” keywords into interesting.dat)

Answers:

Five files (leonardo.jpg, pensive.png, sequin_girl.jpg, spoons.bmp, elder_lady.jpg) appear to have little or no useful metadata.

Three files (bearded_guy.jpg, guy.jpg, and lady_in_tshirt.jpg.jpeg all have very little meta data. They do contain Ducky and Adobe very early in the file – indicating that Photoshop “save for web” option was likely used. Exiftool (with –v3 option or higher) shows that Ducky is an APP12 and Adobe is an APP14 segment. App12 Ducky tags may have Quality, Comment and Copyright information, though these don’t seem to have anything other than quality. (bearded_guy has a quality “K” the other two have quality “P”).

```
bearded_guy.jpg
0000000: ffd8 ffe0 0010 4a46 4946 0001 0200 0064  ......JFIF.....d
0000010: 0064 0000 ffec 0011 4475 636b 7900 0100  .d......Ducky...
0000020: 0400 0000 4b00 00ff ee00 0e41 646f 6265  ....K......Adobe
0000030: 0064 c000 0000 01ff db00 8400 0302 0202  .d..............
```

```
guy.jpg
0000000: ffd8 ffe0 0010 4a46 4946 0001 0200 0064  ......JFIF.....d
0000010: 0064 0000 ffec 0011 4475 636b 7900 0100  .d......Ducky...
0000020: 0400 0000 3c00 00ff ee00 0e41 646f 6265  ....<......Adobe
0000030: 0064 c000 0000 01ff db00 8400 0604 0404  .d..............
```

```
lady_in_tshirt.jpg.jpeg
0000000: ffd8 ffe0 0010 4a46 4946 0001 0200 0064  ......JFIF.....d
0000010: 0064 0000 ffec 0011 4475 636b 7900 0100  .d......Ducky...
0000020: 0400 0000 5000 00ff ee00 2641 646f 6265  ....P.....&Adobe
0000030: 0064 c000 0000 0103 0015 0403 060a 0d00  .d..............
```

One file (girl_with_glasses.jpg) seems to be created using the GD graphics library (http://www.boutell.com/gd or http://www.libgd.org) .

```
2:  FileName = girl_with_glasses.jpg
6:  FileType = JPEG
15:  Comment = CREATOR: gd-jpeg v1.0 (using IJG JPEG v62), quality = 95.
```

The rest all have information that may be useful. Below is a summary based on the fields outlined above in the methodology section.

```
2:  FileName = beach_foot.jpg
6:  FileType = JPEG
17:  | 0)  Make = Canon
18:  | 1)  Model = Canon PowerShot SD550
23:  | 6)  Software = Adobe Photoshop CS3 Windows
43:  | | 14) UserComment =
44:  | | 15) FlashpixVersion = 0100
74:  + [Photoshop directory, 3360 bytes]
81:  | Photoshop_0x042f = ...X.X..........(.....'.7.JPG.
88:  | Photoshop_0x0426 = ?.
91:  | Photoshop_0x03f3 = .
93:  | Photoshop_0x2710 = ..
```

```
94:  | Photoshop_0x03f5 = /ff.lff../ff......2.Z..5.-..
95:  | Photoshop_0x03f8 = ...............................................[snip]
96:  | Photoshop_0x0408 = ..@.@
97:  | Photoshop_0x041e =
98:  | Photoshop_0x041a = ....Real_2.jpg.......null..boundsObjc.Rct1.Top longLef[snip]
99:  | Photoshop_0x0428 = .?.
100:  | Photoshop_0x0414 = .
101:  | PhotoshopThumbnail = ..x...........JFIF..HH...Adobe_CM....Adobed.........[snip]
102:  | Photoshop_0x0421 = ...Adobe Photoshop.Adobe Photoshop CS3.
105:  | | PhotoshopQuality = 6
106:  | | PhotoshopFormat = 0
110:  | XMPToolkit = Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:[snip]
111:  | DateAcquired = 2008:02:24 21:53:16Z
112:  | InstanceID = uuid:37E965F354FFDC11ACB494E4759045DB
113:  | DocumentID = uuid:36E965F354FFDC11ACB494E4759045DB
119:  | Make = Canon
120:  | Model = Canon PowerShot SD550
125:  | CreatorTool = Adobe Photoshop CS3 Windows
127:  | FlashpixVersion = 0100
155:  | ICCProfileName = sRGB IEC61966-2.1
157:  | DerivedFromInstanceID = uuid:278AD8BBFBFCDC119C42E1BC13B20C1D
158:  | DerivedFromDocumentID = uuid:49956B99CFF9DC11BC42E207DC2AA02E
169:  + [ICC_Profile directory with 17 entries]
170:  | ProfileHeader (SubDirectory) -->
172:  | | ProfileCMMType = Lino
173:  | | ProfileVersion = 528
174:  | | ProfileClass = mntr
176:  | | ProfileConnectionSpace = XYZ
177:  | | ProfileDateTime = 1998 2 9 6 49 0
178:  | | ProfileFileSignature = acsp
181:  | | DeviceManufacturer = IEC
182:  | | DeviceModel = sRGB
183:  | | DeviceAttributes = 0 0
186:  | | ProfileCreator = HP
187:  | | ProfileID = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
188:  | 0)  ProfileCopyright = Copyright (c) 1998 Hewlett-Packard Company
189:  | 1)  ProfileDescription = sRGB IEC61966-2.1
195:  | 7)  DeviceMfgDesc = IEC http://www.iec.ch
196:  | 8)  DeviceModelDesc = IEC 61966-2.1 Default RGB colour space – sRGB

2:  FileName = blue_eyes.jpg.jpg
6:  FileType = JPEG
21:  | 4)  Software = Adobe Photoshop CS3 Windows
36:  + [Photoshop directory, 5426 bytes]
43:  | Photoshop_0x042f = ...HH..@...d........'.jpg=
50:  | Photoshop_0x0426 = ?.
53:  | Photoshop_0x03f3 = .
```

```
55:  | Photoshop_0x2710 = ..
56:  | Photoshop_0x03f5 = /ff.lff../ff......2.Z..5.-..
57:  | Photoshop_0x03f8 = ....................................................[snip]
58:  | Photoshop_0x0408 = ..@.@
59:  | Photoshop_0x041e =
60:  | Photoshop_0x041a = ....Angie.......null..boundsObjc.Rct1.Top longLeftlong[snip]
61:  | Photoshop_0x0428 = .?.
62:  | Photoshop_0x0411 = .
63:  | Photoshop_0x0414 = .
64:  | PhotoshopThumbnail = ..x...........JFIF..HH...Adobe_CM....Adobed..........[snip]
65:  | Photoshop_0x0421 = ...Adobe Photoshop.Adobe Photoshop CS3.
68:  | | PhotoshopQuality = 6
69:  | | PhotoshopFormat = 0
73:  | XMPToolkit = Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:[snip]
77:  | CreatorTool = Adobe Photoshop CS3 Windows
81:  | InstanceID = uuid:33054AC9D1F9DC11A61A8FFB545246BA
82:  | DocumentID = uuid:32054AC9D1F9DC11A61A8FFB545246BA
92:  | DerivedFrom (SubDirectory) -->
```

```
2:  FileName = board.jpg
6:  FileType = JPEG
11:  | 0)  Make = CASIO COMPUTER CO.,LTD
12:  | 1)  Model = EX-Z3
17:  | 6)  Software = 1.00
36:  | | 14) MakerNoteCasio2 (SubDirectory) -->
37:  | | + [MakerNotes directory with 35 entries]
42:  | | | 4)  FirmwareDate = 0302280945
73:  | | 15) FlashpixVersion = 0100
```

```
2:  FileName = bride.jpg
6:  FileType = JPEG
21:  | 4)  Software = Adobe Photoshop CS3 Windows
38:  + [Photoshop directory, 7256 bytes]
44:  | | OriginatingProgram = Adobe Photoshop CS3 Windows
46:  | Photoshop_0x042f = ...X.X..........(.....'.llun
53:  | Photoshop_0x0426 = ?.
56:  | Photoshop_0x03f3 = .
58:  | Photoshop_0x2710 = ..
59:  | Photoshop_0x03f5 = /ff.lff../ff......2.Z..5.-..
60:  | Photoshop_0x03f8 = ....................................................[snip]
61:  | Photoshop_0x0400 = .
62:  | Photoshop_0x0402 =
63:  | Photoshop_0x0430 = ..
64:  | Photoshop_0x042d = ..
65:  | Photoshop_0x0408 = ..@.@
66:  | Photoshop_0x041e =
67:  | Photoshop_0x041a = ..b..bride....b...null..boundsObjc.Rct1.Top longLeftlo[snip]
```

```
68:  | Photoshop_0x0428 = .?.
69:  | Photoshop_0x0411 = .
70:  | Photoshop_0x0414 = .
71:  | PhotoshopThumbnail = .t..\...........JFIF..HH...Adobe_CM....Adobed.........[snip]
72:  | Photoshop_0x0421 = ...Adobe Photoshop.Adobe Photoshop CS3.
73:  | Photoshop_0x0fa0 = maniIRFR.8BIMAnDs...null.AFStlongFrInVlLs.Objc.null.Fr[snip]
74:  | Photoshop_0x0fa1 = mfri....
77:  | | PhotoshopQuality = 8
78:  | | PhotoshopFormat = 0
82:  | XMPToolkit = Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:[snip]
84:  | [adding XMP-tiff:Software (1)]
85:  | Software = Adobe Photoshop CS3 Windows
90:  | [adding XMP-xmp:Creatortool]
91:  | Creatortool = Adobe Photoshop CS3 Windows
93:  | CreatorTool = Adobe Photoshop CS3 Windows
96:  | InstanceID = uuid:FBDD9A3559FFDC11A8D0C79FB182F96B
97:  | DocumentID = uuid:FADD9A3559FFDC11A8D0C79FB182F96B
105:  | DerivedFromInstanceID = uuid:F7DD9A3559FFDC11A8D0C79FB182F96B
106:  | DerivedFromDocumentID = uuid:F6DD9A3559FFDC11A8D0C79FB182F96B

2:  FileName = brunette.jpg
6:  FileType = JPEG
11:  | 0)  Make = Canon
12:  | 1)  Model = Canon PowerShot A620
35:  | | 14) MakerNoteCanon (SubDirectory) -->
36:  | | + [MakerNotes directory with 19 entries]
37:  | | | 0)  CanonCameraSettings (SubDirectory) -->
42:  | | | | CanonFlashMode = 0
46:  | | | | CanonImageSize = 0
52:  | | | | CameraISO = 18
56:  | | | | CanonExposureMode = 1
75:  | | | 1)  CanonFocalLength (SubDirectory) -->
81:  | | | 2)  Canon_0x0003 = 0 0 0 0
82:  | | | 3)  CanonShotInfo (SubDirectory) -->
105:  | | | | CameraType = 250
110:  | | | 4)  Canon_0x0000 = 0 0 0 0 0 0
111:  | | | 5)  CanonImageType = IMG:PowerShot A620 JPEG
112:  | | | 6)  CanonFirmwareVersion = Firmware Version 1.00
113:  | | | 7)  FileNumber = 1090933
114:  | | | 8)  OwnerName =
115:  | | | 9)  CanonCameraInfoUnknown32 (SubDirectory) -->
117:  | | | 10) CanonModelID = 24641536
118:  | | | 11) Canon_0x0000 = 0 0 0 0 0 0 0 0 0
119:  | | | 12) CanonAFInfo (SubDirectory) -->
120:  | | | + [SerialData directory, 56 bytes]
123:  | | | | 2)  CanonImageWidth = 3072
124:  | | | | 3)  CanonImageHeight = 2304
```

```
134:  | | | 14) Canon_0x0018 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 [snip]
135:  | | | 15) Canon_0x0019 = 1
136:  | | | 16) DateStampMode = 0
140:  | | | 18) FirmwareRevision = 16777472
141:  | | 15) UserComment =
142:  | | 16) FlashpixVersion = 0100

2:  FileName = curtain_lady.jpg
6:  FileType = JPEG
21:  | 4)  Software = Adobe Photoshop CS3 Windows
36:  + [Photoshop directory, 6542 bytes]
43:  | Photoshop_0x042f = ...X.X..........(.....'.pgpg..
50:  | Photoshop_0x0426 = ?.
53:  | Photoshop_0x03f3 = .
55:  | Photoshop_0x2710 = ..
56:  | Photoshop_0x03f5 = /ff.lff../ff......2.Z..5.-..
57:  | Photoshop_0x03f8 = .....................................................[snip]
58:  | Photoshop_0x0400 = .
59:  | Photoshop_0x0402 =
60:  | Photoshop_0x0430 = ...
61:  | Photoshop_0x042d = ..
62:  | Photoshop_0x0408 = ..@.@
63:  | Photoshop_0x041e =
64:  | Photoshop_0x041a = ..,...curtain_baseline.....,...null..boundsObjc.Rct1.T[snip]
65:  | Photoshop_0x0428 = .?.
66:  | Photoshop_0x0411 = .
67:  | Photoshop_0x0414 = .
68:  | PhotoshopThumbnail = ...... .........JFIF..HH...Adobe_CM....Adobed.........[snip]
69:  | Photoshop_0x0421 = ...Adobe Photoshop.Adobe Photoshop CS3.
70:  | Photoshop_0x0fa0 = maniIRFR.8BIMAnDs...null.AFStlongFrInVlLs.Objc.null.Fr[snip]
71:  | Photoshop_0x0fa1 = mfri....
74:  | | PhotoshopQuality = 6
75:  | | PhotoshopFormat = 0
79:  | XMPToolkit = Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:[snip]
83:  | CreatorTool = Adobe Photoshop CS3 Windows
87:  | InstanceID = uuid:195B43F058FFDC11A8D0C79FB182F96B
88:  | DocumentID = uuid:185B43F058FFDC11A8D0C79FB182F96B
98:  | DerivedFromInstanceID = uuid:A150933A58FFDC11ACB494E4759045DB
99:  | DerivedFromDocumentID = uuid:3FE965F354FFDC11ACB494E4759045DB

2:  FileName = eagle.jpg
6:  FileType = JPEG
17:  | 0)  Make = Canon
18:  | 1)  Model = Canon EOS 350D DIGITAL
40:  | | 13) MakerNoteCanon (SubDirectory) -->
41:  | | + [MakerNotes directory with 24 entries]
42:  | | | Warning = [minor] Adjusted MakerNotes base by 4116
```

```
43:  | | | 0)  CanonCameraSettings (SubDirectory) -->
48:  | | | | CanonFlashMode = 0
52:  | | | | CanonImageSize = 0
58:  | | | | CameraISO = 32767
62:  | | | | CanonExposureMode = 1
81:  | | | 1)  CanonFocalLength (SubDirectory) -->
87:  | | | 2)  Canon_0x0003 = 100 0 0 0
88:  | | | 3)  CanonShotInfo (SubDirectory) -->
110:  | | | | CameraType = 252
115:  | | | 4)  CanonImageType = Canon EOS 350D DIGITAL
116:  | | | 5)  CanonFirmwareVersion = Firmware 1.0.1
117:  | | | 6)  OwnerName = unknown
118:  | | | 7)  SerialNumber = 330108222
119:  | | | 8)  CanonCameraInfoUnknown (SubDirectory) -->
122:  | | | + [CanonCustom directory with 9 entries]
132:  | | | 10) CanonModelID = 2147484041
133:  | | | 11) CanonAFInfo (SubDirectory) -->
134:  | | | + [SerialData directory, 48 bytes]
137:  | | | | 2)  CanonImageWidth = 3456
138:  | | | | 3)  CanonImageHeight = 2304
147:  | | | 13) SerialNumberFormat = 2684354560
148:  | | | 14) Canon_0x0019 = 1
150:  | | | 16) CanonFileInfo (SubDirectory) -->
152:  | | | | FileNumber = 14229128
178:  | | | 18) Canon_0x00aa = 10 244 1024 1024 1048
214:  | | | 22) Canon_0x4002 = 5352 0 0 0 5657 65535 65280 41984 1297 65535 400 0 286[snip]
219:  | | 14) UserComment =
220:  | | 15) FlashpixVersion = 0100
244:  | About = uuid:faf5bdd5-ba3d-11da-ad31-d33d75182f1b
259:  | [adding XMP-tiff:Make (1)]
260:  | Make = Canon
261:  | [adding XMP-tiff:Model (1)]
262:  | Model = Canon EOS 350D DIGITAL

2:  FileName = girl.jpg.jpg
6:  FileType = JPEG
21:  | 4)  Software = Adobe Photoshop CS3 Windows
36:  + [Photoshop directory, 6470 bytes]
43:  | Photoshop_0x042f = ...HH..@...d........'.llun.
50:  | Photoshop_0x0426 = ?.
53:  | Photoshop_0x03f3 = .
55:  | Photoshop_0x2710 = ..
56:  | Photoshop_0x03f5 = /ff.lff../ff......2.Z..5.-..
57:  | Photoshop_0x03f8 = ....................................................[snip]
58:  | Photoshop_0x0408 = ..@.@
59:  | Photoshop_0x041e =
60:  | Photoshop_0x041a = ....pageant-makeover-lg.......null..boundsObjc.Rct1.To[snip]
```

```
61:  | Photoshop_0x0428 = .?.
62:  | Photoshop_0x0411 = .
63:  | Photoshop_0x0414 = .
64:  | PhotoshopThumbnail = .x..h..........JFIF..HH...Adobe_CM....Adobed..........[snip]
65:  | Photoshop_0x0421 = ...Adobe Photoshop.Adobe Photoshop CS3.
68:  | | PhotoshopQuality = 6
69:  | | PhotoshopFormat = 1
73:  | XMPToolkit = Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:[snip]
77:  | CreatorTool = Adobe Photoshop CS3 Windows
81:  | InstanceID = uuid:241DC1DD6BFBDC11B447D95B2AB04FF4

2:  FileName = keyboard.jpg
6:  FileType = JPEG
17:  | 0)  Make = Canon
18:  | 1)  Model = Canon PowerShot G7
42:  | | 13) MakerNoteCanon (SubDirectory) -->
43:  | | + [MakerNotes directory with 26 entries]
44:  | | | Warning = [minor] Adjusted MakerNotes base by 4148
45:  | | | 0)  CanonCameraSettings (SubDirectory) -->
50:  | | | | CanonFlashMode = 0
54:  | | | | CanonImageSize = 0
60:  | | | | CameraISO = 16584
64:  | | | | CanonExposureMode = 1
83:  | | | 1)  CanonFocalLength (SubDirectory) -->
89:  | | | 2)  Canon_0x0003 = 0 0 0 0
90:  | | | 3)  CanonShotInfo (SubDirectory) -->
113:  | | | | CameraType = 250
118:  | | | 4)  Canon_0x0000 = 0 0 0 0 0 0
119:  | | | 5)  CanonImageType = IMG:PowerShot G7 JPEG
120:  | | | 6)  CanonFirmwareVersion = Firmware Version 1.00
121:  | | | 7)  FileNumber = 1051758
122:  | | | 8)  OwnerName =
123:  | | | 9)  CanonCameraInfoPowerShot (SubDirectory) -->
129:  | | | 10) CanonModelID = 26738688
130:  | | | 11) Canon_0x0000 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
131:  | | | 12) CanonAFInfo2 (SubDirectory) -->
132:  | | | + [SerialData directory, 96 bytes]
137:  | | | | 4)  CanonImageWidth = 3648
138:  | | | | 5)  CanonImageHeight = 2736
146:  | | | | 13) Canon_AFInfo2_0x000d = 0 0
149:  | | | 14) Canon_0x0018 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 [snip]
150:  | | | 15) Canon_0x0019 = 1
151:  | | | 16) DateStampMode = 0
155:  | | | 18) FirmwareRevision = 16779008
156:  | | | 19) Canon_0x001f = 138 1 0 4 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 [snip]
157:  | | | 20) Canon_0x0022 = 416 0 0 16 8 24 16 640 480 65216 65302 320 234 0 8 384[snip]
176:  | | | 24) Canon_0x0027 = 4 0
```

```
177:  | | | 25) Canon_0x0028 = 96 174 151 184 252 205 251 238 66 11 198 135 121 95 250 92
178:  | | 14) UserComment =
179:  | | 15) FlashpixVersion = 0100
207:  | About = uuid:faf5bdd5-ba3d-11da-ad31-d33d75182f1b
220:  | [adding XMP-tiff:Make (1)]
221:  | Make = Canon
222:  | [adding XMP-tiff:Model (1)]
223:  | Model = Canon PowerShot G7

2:  FileName = tulips.jpg
6:  FileType = JPEG
11:  | 0)  Make = Canon
12:  | 1)  Model = Canon PowerShot A520
35:  | | 14) MakerNoteCanon (SubDirectory) -->
36:  | | + [MakerNotes directory with 17 entries]
37:  | | | 0)  CanonCameraSettings (SubDirectory) -->
42:  | | | | CanonFlashMode = 0
46:  | | | | CanonImageSize = 0
52:  | | | | CameraISO = 16
56:  | | | | CanonExposureMode = 1
75:  | | | 1)  CanonFocalLength (SubDirectory) -->
81:  | | | 2)  Canon_0x0003 = 0 0 0 0
82:  | | | 3)  CanonShotInfo (SubDirectory) -->
105:  | | | | CameraType = 250
110:  | | | 4)  Canon_0x0000 = 0 0 0 0 0 0
111:  | | | 5)  Canon_0x0000 = 0 0 0 0
112:  | | | 6)  CanonAFInfo (SubDirectory) -->
113:  | | | + [SerialData directory, 56 bytes]
116:  | | | | 2)  CanonImageWidth = 2272
117:  | | | | 3)  CanonImageHeight = 1704
127:  | | | 8)  CanonImageType = IMG:PowerShot A520 JPEG
128:  | | | 9)  CanonFirmwareVersion = Firmware Version 1.00
129:  | | | 10) FileNumber = 1050578
130:  | | | 11) OwnerName =
131:  | | | 12) CanonModelID = 22413312
132:  | | | 13) Canon_0x0018 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 [snip]
133:  | | | 14) Canon_0x0019 = 1
134:  | | | 15) DateStampMode = 0
135:  | | | 16) CanonCameraInfoUnknown16 (SubDirectory) -->
137:  | | 15) UserComment =
138:  | | 16) FlashpixVersion = 0100

2:  FileName = veiled_lady.jpg
6:  FileType = JPEG
21:  | 4)  Software = Adobe Photoshop CS3 Windows
36:  + [Photoshop directory, 5372 bytes]
43:  | Photoshop_0x042f = Z.HH..@...d........'.l.jpgj
```

```
50:  | Photoshop_0x0426 = ?.
53:  | Photoshop_0x03f3 = .
55:  | Photoshop_0x2710 = ..
56:  | Photoshop_0x03f5 = /ff.lff../ff......2.Z..5.-..
57:  | Photoshop_0x03f8 = .....................................................[snip]
58:  | Photoshop_0x0408 = ..@.@
59:  | Photoshop_0x041e =
60:  | Photoshop_0x041a = ......aal.........null..boundsObjc.Rct1.Top longLeftlo[snip]
61:  | Photoshop_0x0428 = .?.
62:  | Photoshop_0x0411 = .
63:  | Photoshop_0x0414 = .
64:  | PhotoshopThumbnail = .q..T..........JFIF..HH...Adobe_CM....Adobed.........[snip]
65:  | Photoshop_0x0421 = ...Adobe Photoshop.Adobe Photoshop CS3.
68:  | | PhotoshopQuality = 6
69:  | | PhotoshopFormat = 1
73:  | XMPToolkit = Adobe XMP Core 4.1-c036 46.276720, Mon Feb 19 2007 22:[snip]
77:  | CreatorTool = Adobe Photoshop CS3 Windows
81:  | InstanceID = uuid:C853B1E86935DC11B00FA7B1504A2B4F
```

Tools used:
Type: command line meta-data inspection utility
Name: exiftool
Publisher: Phil Harvey
Open Source (Perl License)
Site: www.sno.phy.queensu.ca/~phil/exiftool

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

104 Signature Analysis

Methodology:

Similar to challenge 102, this task can be performed using dedicated forensics tools or via readily available open source tools. Pro-Discover, EnCase and FTK all do this natively.

A sample text log from EnCase version 5 is provided below. Gary Kessler maintains a file type list at http://www.garykessler.net/library/file_sigs.html which could be used as a basis for creating a new tool, or as an input data to a modular tool.

Most unix based Operating Systems contain the “file” utility which analyzes various portions of files to determine their type (unix applications do not make use of file extensions such as .doc nearly as much as Windows applications). Cursory inspection of the “file” output reveals several files that may not match their file extension (**bolded** below). An investigator may not be familiar with some of the less common file types such as 1ST or CB files, and not know what to expect for “file” output.

Finally, Harlan Carvey created a tool called sigs. At some point Harlan wrote a book and now sigs is distributed with the book (and not as easy to find online). Since this was originally written in perl, you’ll need the perl2exe dll files in order to actually run the exe file. It is recommended that you obtain a Helix forensics LiveCD and use sigs.exe and p2x580.dll found in the windows portion of that CD ROM. You can obtain headersig.txt from http://www.filesig.co.uk/. This results in a completely free solution, output of sigs can be found below – and even though headersig.txt is a bit out of date, it can identify matches fairly well. Mismatches are in **bold**.

Another Open Source utility is trid, available at http://mark0.net/soft-trid-e.html. To use this program you need the defs (definitions) and the program itself. Both available from the website. (see below for instructions on download and running using Cygwin on Windows XPSP2). This program assigned a probability value along with the detected file type.

Encase Search Output (abbreviated):

```
Searching
Status: Completed
Start: 05/06/08 23:21:38
Stop: 05/06/08 23:21:39
Time: 0:00:01
Files: 21
Files scanned: 20
```

Signature mismatches: 5 (this indicates that encase believes 5 files have the wrong extension)
Hash values: 20
1) dc3-2008\Single Files\245.JPG
File Type JPEG
File Category Picture
Signature Match

2) dc3-2008\Single Files\249.JPG
File Type JPEG
File Category Picture
Signature Match

3) dc3-2008\Single Files\255.JPG
File Type JPEG
File Category Picture
Signature Match

4) dc3-2008\Single Files\AutoWire.bmp
File Type Bitmap Image
File Category Picture
Signature Match

5) dc3-2008\Single Files\blank.jpg
File Type JPEG
File Category Picture
Signature ! Bad signature

6) dc3-2008\Single Files\blue.bmp
File Type Bitmap Image
File Category Picture
Signature ! Bad signature

7) dc3-2008\Single Files\Bluestar.gif
File Type GIF
File Category Picture
Signature Match

8) dc3-2008\Single Files\Chaff_Floral_1179.bmp
File Type Bitmap Image
File Category Picture
Signature Match

9) dc3-2008\Single Files\Chaff_Landscape_158.gif

File Type	GIF
File Category	Picture
Signature	Match

10) dc3-2008\Single Files\Chaff_Landscape_161.gif

File Type	GIF
File Category	Picture
Signature	Match

11) dc3-2008\Single Files\CLOCK.MOV

File Type	**QuickTime Movie**
File Category	**Multimedia**
Signature	*** MS Compressed**

12) dc3-2008\Single Files\DollL Sales Worldwide.html

File Type	**Web Page**
File Category	**Picture**
Signature	*** JPEG Image**

13) dc3-2008\Single Files\intro.mpeg

File Type	**MPEG Video**
File Category	**Multimedia**
Signature	*** ZIP Compressed**

14) dc3-2008\Single Files\ipp_0004.asp

File Type	Adobe Photoshop Separation Tables
File Category	Code\Application
Signature	Match

15) dc3-2008\Single Files\pctools.zip

File Type	**ZIP Compressed**
File Category	**Archive**
Signature	**! Bad signature**

16) dc3-2008\Single Files\SAILBOAT.JPG

File Type	**JPEG**
File Category	**Picture**
Signature	*** Compiled HTML Help File**

17) dc3-2008\Single Files\straightline.tif

File Type	**TIFF Image**
File Category	**Picture**
Signature	**! Bad signature**

18) dc3-2008\Single Files\SYSTEM.1ST

File Type	**Readme**
File Category	**Document**
Signature	*** Windows 95 Registry Files**

19) dc3-2008\Single Files\SYSTEM.CB

File Type	MS Clean Boot
File Category	Windows
Signature	Match

20) dc3-2008\Single Files\Windows.wav

File Type	**Waveform Audio**
File Category	**Multimedia\Sound**
Signature	**! Bad signature**

‘File’ output:

```
~>file ./*
./245.JPG:                JPEG image data, EXIF standard
 ./249.JPG:                JPEG image data, JFIF standard 1.01
 ./255.JPG:                JPEG image data, EXIF standard
 ./AutoWire.bmp:             PC bitmap data, Windows 3.x format, 601 x 440 x 8
./blank.jpg:               ASCII English text, with CRLF line terminators
./blue.bmp:                 ASCII English text, with CRLF line terminators
 ./Bluestar.gif:            GIF image data, version 89a, 1002 x 635
 ./Chaff_Floral_1179.bmp:     PC bitmap data, Windows 3.x format, 1024 x 768 x 8
 ./Chaff_Landscape_158.gif:   GIF image data, version 87a, 252 x 204
 ./Chaff_Landscape_161.gif:   GIF image data, version 89a, 200 x 132
./CLOCK.MOV:                Microsoft Cabinet archive data, 40411 bytes, 2 files
./DollL Sales Worldwide.html: JPEG image data, JFIF standard 1.01
./intro.mpeg:               Zip archive data, at least v2.0 to extract
 ./ipp_0004.asp:            ASCII English text, with CRLF line terminators
./pctools.zip:             data
./SAILBOAT.JPG:             MS Windows HtmlHelp Data
./straightline.tif:         ASCII English text, with CRLF line terminators
./SYSTEM.1ST:               Windows 95/98/ME registry file
```

```
./SYSTEM.CB:             ASCII text, with CRLF line terminators
./Windows.wav:             shell archive or script for antique kernel text
```

Sigs output:

```
>\245.JPG, Sig not listed. (JPG,FFD8FFE12C)
>\249.JPG, Sig match.
>\255.JPG, Sig not listed. (JPG,FFD8FFE12B)
>\AutoWire.bmp, Sig match.
>\blank.jpg, Sig not listed. (jpg,3C25402043)
>\blue.bmp, Sig not listed. (bmp,3C25402043)
>\Bluestar.gif, Sig match.
>\Chaff_Floral_1179.bmp, Sig match.
>\Chaff_Landscape_158.gif, Sig match.
>\Chaff_Landscape_161.gif, Sig match.
>\CLOCK.MOV, Sig does not match. (MOV,4D53434600)
>\DollL Sales Worldwide.html, Sig does not match. (html,FFD8FFE000)
>\intro.mpeg, Sig does not match. (mpeg,504B030414)
>\ipp_0004.asp, Sig not listed. (asp,3C25402043)
>\pctools.zip, Sig not listed. (zip,3082211806)
>\SAILBOAT.JPG, Sig not listed. (JPG,4954534603)
>\straightline.tif, Sig not listed. (tif,3C25402043)
>\SYSTEM.1ST, Sig not listed. (1ST,4352454700)
>\SYSTEM.CB, Sig not listed. (CB,5B4B657962)
>\Windows.wav, Sig does not match. (wav,3A42617365)
```

Trid installation and output:

```
$ wget http://mark0.net/download/triddefs.zip
$ wget http://mark0.net/download/trid_linux.zip
$ unzip ./*.zip
$ ./trid.exe 104_Signature_Analysis_Challenge2008/*

TrID/32 - File Identifier v2.02 - (C) 2003-06 By M.Pontello
Definitions found:  3273
Analyzing...

File: 104_Signature_Analysis_Challenge2008/245.JPG
 48.6% (.JPG) JFIF-EXIF JPEG Bitmap (5000/1/1)

File: 104_Signature_Analysis_Challenge2008/249.JPG
```

```
 32.2% (.JPG) JFIF-EXIF JPEG Bitmap (5000/1/1)

File: 104_Signature_Analysis_Challenge2008/255.JPG
 48.6% (.JPG) JFIF-EXIF JPEG Bitmap (5000/1/1)

File: 104_Signature_Analysis_Challenge2008/AutoWire.bmp
100.0% (.BMP) Windows Bitmap (2000/1)

File: 104_Signature_Analysis_Challenge2008/Bluestar.gif
 60.0% (.GIF) GIF89a Bitmap (6000/1)

File: 104_Signature_Analysis_Challenge2008/CLOCK.MOV
 99.9% (.CAB) Microsoft Cabinet Archive (8000/1)

File: 104_Signature_Analysis_Challenge2008/Chaff_Floral_1179.bmp
 50.6% (.RLE) Run Lenght Encoded bitmap (2050/26)

File: 104_Signature_Analysis_Challenge2008/Chaff_Landscape_158.gif
 60.0% (.GIF) GIF87a Bitmap (6001/2)

File: 104_Signature_Analysis_Challenge2008/Chaff_Landscape_161.gif
 60.0% (.GIF) GIF89a Bitmap (6000/1)

File: 104_Signature_Analysis_Challenge2008/DollL Sales Worldwide.html
 32.2% (.JPG) JFIF-EXIF JPEG Bitmap (5000/1/1)

File: 104_Signature_Analysis_Challenge2008/SAILBOAT.JPG
 81.0% (.CHI) Windows HELP Index (17144/6)

File: 104_Signature_Analysis_Challenge2008/SYSTEM.1ST
100.0% (.DAT) Windows 9x Registry Hive (4005/2)

File: 104_Signature_Analysis_Challenge2008/SYSTEM.CB
100.0% (.INI) Generic INI configuration (1000/1)

File: 104_Signature_Analysis_Challenge2008/Windows.wav
100.0% (.CNT) Help File Contents (3000/1/1)

File: 104_Signature_Analysis_Challenge2008/blank.jpg
100.0% (.HTML) HyperText Markup Language (3000/1/1)
```

```
File: 104_Signature_Analysis_Challenge2008/blue.bmp
    Unknown!
```

File: 104_Signature_Analysis_Challenge2008/intro.mpeg
100.0% (.ZIP) ZIP compressed archive (4000/1)

File: 104_Signature_Analysis_Challenge2008/ipp_0004.asp
100.0% (.HTML) HyperText Markup Language (3000/1/1)

File: 104_Signature_Analysis_Challenge2008/pctools.zip
49.9% (.DER) DER encoded X509 Certificate (2000/1)

File: 104_Signature_Analysis_Challenge2008/straightline.tif
100.0% (.HTML) HyperText Markup Language (3000/1/1)

Answers:

CLOCK.MOV should be a .CAB file.

This file can be opened with 7-zip as a .cab (cabinet) file. The .cab file contains one text file deleteme.txt and a dll file. These files were not explored further. The extracted files can be found in the CLOCK-extracted directory.

DollL Sales Worldwide.html should be a .jpg file.

This is a picture of a sailboat (possibly originally SAILBOAT.JPG?) that seems to be taken from behind a tree. Based on exif data, it was taken on 2/18/2008 at 4:12 PM by a Canon Powershot SD850 IS. In conjunction with intro.mpeg, one might presume that the original filename on the computer was about 262.JPG (based on file-number). The original File number was 100-0189, it was imported with Windows Photo Gallery 6.0.6000.16386.

SAILBOAT.JPG should be a .chm file.

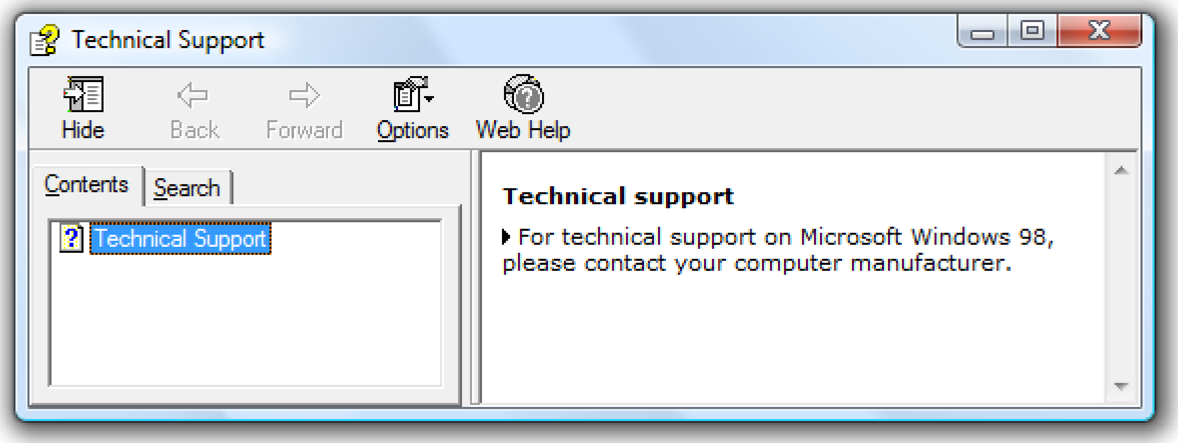
Mostly due to the magic signature of ITSF, this was originally thought to be a windows help index (.chi) file that is typically found on Read Only CDROMs used to speed up access to files. Essentially an index for CHMs. This chm file was created with HHA version 4.72.8033. The current version of HHA as used in the HTML Help Workshop distributed by Microsoft is 4.74.8702.0. This seems to be a generic help file from a Windows 98 machine – but this has not been verified.

The sole “Contents” item is: “Technical Support” which has the following htm associated with it:

Technical Support

```
>For technical support on Microsoft Windows 98, please contact your
```

```
computer manufacturer.
```



Technical support

For technical support on Microsoft Windows 98, please contact your computer manufacturer.

http://go.microsoft.com/fwlink/?linkid=14188 (help workshop)
http://go.microsoft.com/fwlink/?linkid=14581 (docs).

SYSTEM.1ST should a .dat file.

This is the older win95/98 style registry file format. Newer versions of Windows have issues utilizing data in this file format. If viewing is all that is desired, I recommend using a parsing library like Parse::Win32Registry for perl. Using dumpreg.pl (see appendix, and CDROM) it is easy to see the root keys are:

```
        Software
        System
        Enum
        Hardware
        Network
```

Many other subkeys are also present, but including them here would result in a MUCH larger report. If this is of interest I recommend running dumpreg.pl. Note: you must first install the library:

```
#>cpan
cpan>install Parse::Win32Registry
cpan>exit
#>./dumpreg.pl -ri SYSTEM.dat
```

SYSTEM.CB should be a .ini file.

ini files are in many ways similar to conf or cfg files in unix based systems. They are plain text and used to specify options for something. This file contains:

```
        [Keyboard]
        layout=kbdus.kbd
```

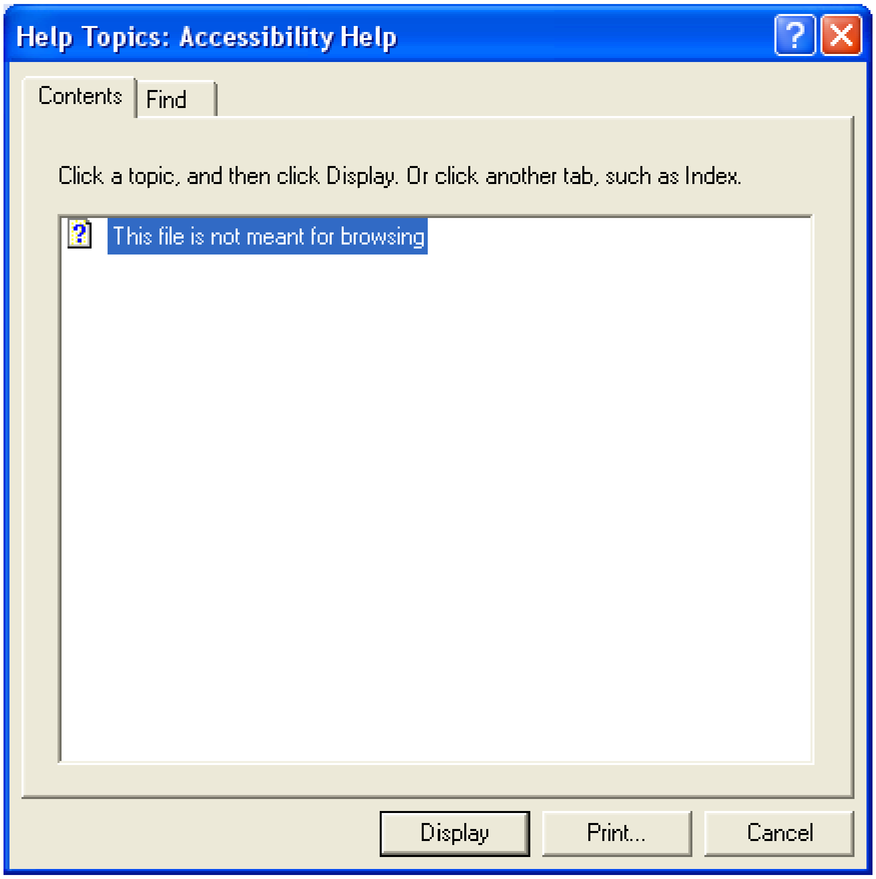
```
[boot]
*DisplayFallback=0

[Intl]
ACP=1252
OEMCP=437
SystemLocale=0000040
```

Windows.wav should be a .cnt file.

.cnt files are associated with windows .hlp files. This particular one should accompany Access.hlp. The original name of this file was likely Access.cnt. Older, non-compiled (.chm) help files often are paired with .cnt files that describe members of the help set. The :Title is the tile to be displayed, the following lines dictate what is displayed in the contents tab when the .hlp file opened. The leading number is the 'heading level', the next string is the text to display. When present the equals sign (=) and string to the right create a hyperlink displaying the display text and linking to the string to the right of the =.

The contents of Windows.cnt:
:Base Access.hlp
:Title Accessibility Help
1 This file is not meant for browsing=WIN_HELP_AUTOCLOSE

When opening Access.hlp, in the contents tab you would see:



Click a topic, and then click Display. Or click another tab, such as Index.

This file is not meant for browsing

blank.jpg should be a .asp file.

There is a fair amount of html at the end of this document, but the document has man plaintext comments and visual basic code that allow it do be identified as part of the Microsoft Internet Printing Project. Internet printing can be enabled by Windows 2000 or later as part of IIS. This particular document was likely ipp_0001.asp, which was probably accessible at
ttp://servername%/printers/ipp_0001.asp
See http://support.microsoft.com/kb/313058 and/or install IPP under IIS on a test machine.

intro.mpeg should be a .zip file.

This file can be opened with 7-zip. It contains 10 JPG files. These files can be found in the intro-extracted directory. Based on exif data these pictures were taken with Canon cameras (SD550 and SD850 IS) on Feb. 2, 26, and 20 in 2008. The pictures seem to be of some sort of vacation resort. The original fild numbers for the SD550 are 113-3658 through 113-3660; for the SD850 : 100-0022, and 100-0036 through 100-0041.

pctools.zip should be a .p7b file.

Simple strings or hex view shows that "verisign" occurs repeatedly. Guessing the .der extension an using openssl the asn1 records can be parsed:

```
~>openssl asn1parse -in pctools.der -inform DER
```

This reveals that this file is pkcs7-signed, it uses md5/sha1. md5WithRSAEncryption, rsaEncryption. There are several references to Microsoft Corp, and Microsoft Root Authority, Redmond, Washington, US. Microsoft windows Hardware Compatibility PCA (or PUBLISHER). There may also be an X509v3 cert embedded.

the openssl output lead me to research Microsoft windows's built in certificate handling software. If the file is name .p7b, then the built in certmng certificate tool (XP and Vista tested) show that there are actually 5 certs embedded. A summary can be found in pctools-exported.txt. the certmng tool can be used to export each of the 5 to der or base64 encoded files. The 5 exported certs can be found in the pctools-exported directory, and can be viewed on a windows machine by double-clicking on them.

straightline.tif should be a .asp file.

See the summary for blank.jpg. This is another IPP file, likely originally ipp_0002.asp.

Tools used:
Type: Hex editor
Name: 010 editor
Publisher: sweetscape software
Commercial
Site: www.010editor.com

Type: Foresic specialty software
Name: Encase v5
Publisher: guidance software
Commercial
Site: www.guidancesoftware.com

Type: Help editing utility
Name: HTML Help Workshop
Publisher: Microsoft Corp
Commercial (free)
Site: http://go.microsoft.com/fwlink/?linkid=14188

Type: utility
Name: file
Publisher: GNU / Fedora project
Open Source
Site: www.gnu.org/binutils , fedora.org

Type: specialty signature software
Name: TrID
Publisher: Marco Pontello
Compiled (free for non-commercial use)
Site: http://mark0.net/soft-trid-e.html

Type: perl module
Name: Parse-Win32Registry
Publisher: James Macfarlane
Open Source
Site: http://search.cpan.org/~jmacfarla/Parse-Win32Registry-0.40/lib/Parse/Win32Registry.pm

Type: Forensics LiveCD
Name: helix
Publisher: e-defense
Much is OpenSource (see the CD ROM for more licensing information)
Site: http://www.e-fense.com/helix/

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

201 – Missing Header Reconstruction

Methodology:

Exiftool can quickly show some of the file portions that contain errors by limiting the output to warnings:

```
#>exiftool ./*.jpg | grep warning
```

The file command predicts that Chaff_landscape_272.GIF is actually a JPG…in fact, all the techniques used in 104 can be used to aide in reconstruction.

Unfortunately however, much of this challenge was performed “manually.” By studying the BMP, JPG, and GIF file formats, then loading each file into a hex editor, we inspected the files looking for areas of data that looked like they belonged to a particular file type. Instead of looking for the file header, known structures or ranges were sought out further into the file. For example a gif file has a file header, but then also has structures like the Logical Screen Descriptor, Global and Local Color Tables, even the picture data may be broken down into Application Extensions, ImageDescriptors, Image Data, etc. A file of interest can be scanned byte by byte looking for matches. For example, the Logical Screen Descriptor can be described as:

```
short width;
short height;
1 bit hasGlobalColorTable;
3 bits Resolution;
1 bit isGlobalTableSorted
3 bits sizeOfGlobalTable;
1 byte BackgroundColorIndex
1 byte pixelAspectRatio
```

when scanning through the files, obscene width to height relationships (1 pixel higher 5000 pixels wide, anything > 10000, etc) can be ignored, the sort flag is only valid in 89a (not 87a) files, and only relates to the GlobalColorTable, having this bit set doesn't make sense unless the hasGlobalColorTable bit is set, etc. This assumes that records other than the Logical Screen Descriptor were damaged. Turning our attention to repairing, assuming that enough other records were found and the Logical Screen Descriptor was damaged. Knowing how the fields are used is paramount to a successful fix. Some fields lend themselves to recovery techniques, for example the height and width are actually just a minimum resolution required to display the image on a hardware device (if the device isn't capable of the resolution the image is scaled). From a recover point of view this is good because the values just need to be set high enough for the image to display, plus the width and height can likely be determined from an Image Descriptor found later in the file. Other repairs are slightly more difficult, for example the size of the GlobalColorTable. If this is corrupted the actual size of the table must be calculated somehow. Other records in the GIF format start with separators, such as byte 0x21 or 0x2c. These records can be located using similar techniques to the Logical Screen Descriptor above. The GlobalColorTable must be contiguous (so prior to any other separator) and the size must be a power of 2 (well a power of 2 times 3 bytes because each 'entrie' is a 3byte RGB value, so a 256 entry table would take up 768 bytes). This knowledge can be used to size up the space between a Logical Screen Descriptor and the first other record type located.

The only good way to report alterations to these binary files is to compare the hex representations of the original and the repaired file. In each case below the original file name is listed, the original bytes are listed and the repaired bytes are listed. Note that the repair of Chaff_Landscape_158 results in the size of the file changing which means that from the point of the change until the end of file is “no longer aligned” which results in a large diff. For this reason the diff is not listed here.

```
110.JPG
< 0000000: 0000 0000 0000 0000 0000 0000 4d4d 002a  ............MM.*
---
> 0000000: ffd8 ffe1 2fb6 4578 6966 0000 4d4d 002a  ..../.Exif..MM.*

112.JPG
< 0000000: 4f70 656e 2e6d 652e 2e2e 0000 4d4d 002a  Open.me.....MM.*
---
> 0000000: ffd8 ffe1 3036 4578 6966 0000 4d4d 002a  ....06Exif..MM.*

226.JPG
< 0000000: 4a46 4946 2e2e ffd8 ffe0 0000 4d4d 002a  JFIF........MM.*
---
> 0000000: ffd8 ffe1 350e 4578 6966 0000 4d4d 002a  ....5.Exif..MM.*

Chaff_Buildings_625.gif
< 0000000: 0000 0000 0000 0000 7b02 e600 00b1 cef0  ........{.......
---
> 0000000: 4749 4638 3961 ea03 7b02 e600 00b1 cef0  GIF89a..{.......

Chaff_Floral_1179.gif
< 0000000: 4749 4638 3961 a000 df00 3604 0000 2800  GIF89a....6...(.
---
> 0000000: 424d 0000 1000 a000 df00 3604 0000 2800  BM........6...(.

Chaff_Landscap_219.bmp
< 0000000: 424d 3604 0c00 0000 0000 f700 0000 0000  BM6.............
---
> 0000000: 4749 4638 3961 a000 df00 f700 0000 0000  GIF89a..........

Chaff_Landscap_272.GIF
< 0000000: ffd8 2e44 fa64 c82e 7a00 f53f 00ff ffff  ...D.d..z..?....
---
```

```
> 0000000: 4749 4638 3961 a000 7a00 f53f 00ff ffff  GIF89a..z..?....
```

Some reference sites include:
http://www.tinaja.com/glib/expbmp.pdf
http://local.wasp.uwa.edu.au/~pbourke/dataformats/bmp/
http://www.fileformat.info/format/gif/#GIF-DMYID.2

Answers:

All images were repaired and can be found in the 201 directory of the media that accompanied this report. Thumbnails and a short description can be found here.

OK_110.JPG

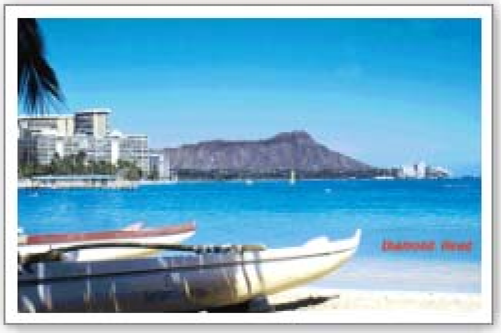
A plate of food

OK_112.JPG

Two turtles on rocks

OK_226.JPG

Coconuts



OK_Chaff_Buildings_625.gif

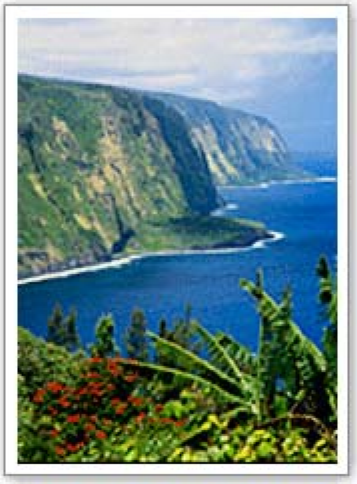
Sailboat by a bay (also tall buildings)
Text: Diamond Head

OK_Chaff_Floral_1179.gif.bmp

Pink flowers

OK_Chaff_Landscape_158.gif

Fall landscape near a river



OK_Chaff_Landscape_219.bmp.gif

Island coast with tall bluffs

OK_Chaff_Landscape_272.GIF

Birds near shoreline rocks

Tools used:
Type: Hex editor
Name: 010 editor
Publisher: sweetscape software
Commercial
Site: www.010editor.com

In particular the “templates” feature. See http://www.sweetscape.com/010editor/templates/

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

202 – Password Recovery Challenge

Methodology:

Given the experience gained from challenges 303 and 403, we briefly entertained the idea of writing a custom attack for this challenge. However, discovering that POI (see 303 for more information on POI) did not support passwords for Word documents at all disheartened us and we resorted to using COTS products. Elcomsoft's Distributed Password Recovery software made short work of this challenge using only one machine (a machine with 8 cores to dedicate). The eight cores averaged about 22 million trys per second and recovered the keys in a matter of hours. Usage is very simple. Add a new job (select the word file you are interested in), select "key" method, then click on the "start" button.

Note that a key attack does allow decryption of the documents, but does not allow the original password to be known. A password based attack could possibly take much, much longer. If knowing the original password is desirable an FPGA or GPU based attack is recommended. Also, since the keyspace is known (and relatively small), solutions such as "rainbow tables" do exist and would theoretically be able to determine the correct key in minutes.

Answers:

Filename	Key	Time to find key
fouo.doc	4TH5Q7U2BCAJF	1 hour 37 minutes
confidential.doc	HDAJX32AZSKDB	11 hours 5 minutes
secret.doc	F1G4469DEFD6H	9 hours 2 minutes

fouo.doc is a four page (1802 words) article titled "Bigfoot — a contemporary belief legend" by Joyce Bynum.

confidential.doc is a six page (3118 words) article titled "Living Ape-Men" which is from Forbidden Archeology: The Hidden History of the Human Race by Michael Cremo.

Secret.doc is an eight page (4055 words) article titled "Sasquatch and Scientists: Reporting Scientific Anomalies" by Ron Westrum .

The three decrypted files can be found the "decrypted" directory for challenge 202 on the media that accompanies this report.

Tools used:
Type: dedicated password cracking software
Name: Distributed Password Recovery
Publisher: Elcomsoft
Commercial
Site: http://www.elcomsoft.com/edpr.html?r1=pr&r2=wpa

Note: Since only one machine was used, Advanced Office Password Breaker would likely have worked just as well. We did not have access to this product.

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

203 Registry Analysis

Methodology:

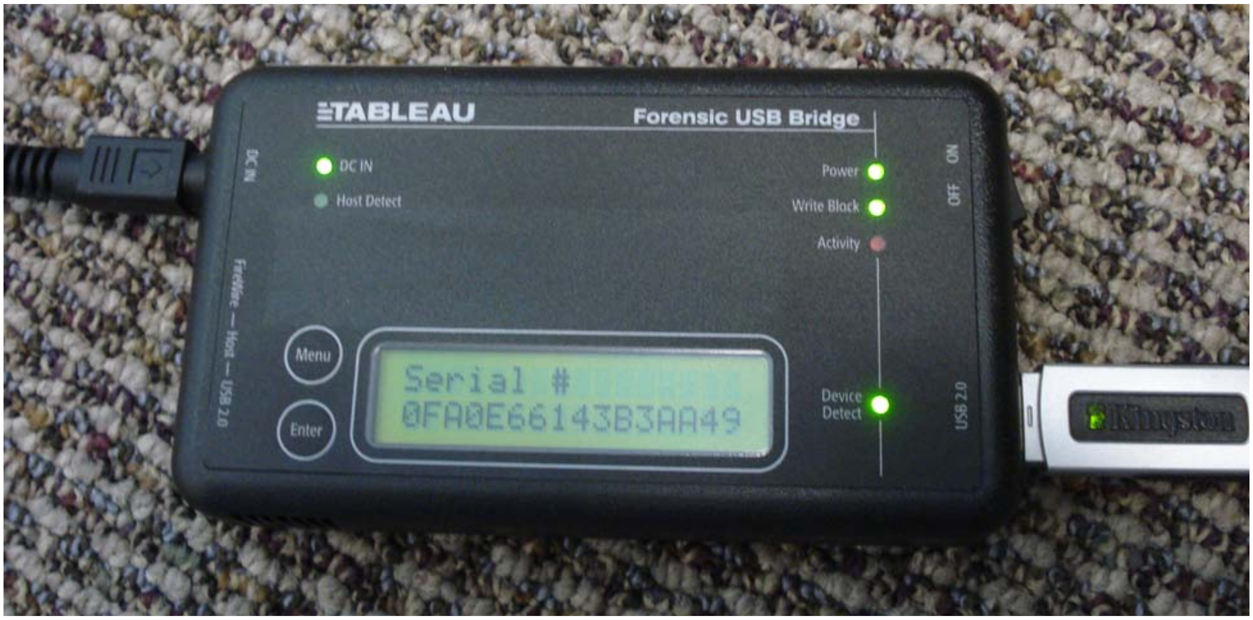
The provided “SYSTEM” file is essentially the contents that would be found under “HKEY_LOCAL_MACHINE” when observing the system registry via regedit. This will be abbreviated as HKLM for the rest of this section.

HKLM\System\ContralSet00x\Enum\USBSTOR contains information about USB devices that have been attached to the system. This particular machine has these entries related to USB disk devices. The created date for the keys are also noted below (all in 2008).

Disk&Ven_Flash&Prod_Drive_AU_USB20&Rev_8.07\BQENV5CR&0	4/1	21:07:32
ParentIdPrefix 8&252733d5&0		
Disk&Ven_SAMSUNG&Prod_HD501LJ&Rev_0-10\152D203380B6&0	4/1	21:07:32
ParentIdPrefix ????????????		
Disk&Ven_USB_2.0&Prod_Flash_Disk&Rev_1100\A127000000000063&0	4/1	21:07:32
ParentIdPrefix 8&6ae132f&0		
Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102C0441&0	4/1	21:07:32
ParentIdPrefix 8&1b050223&0		
Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102B028C&0	4/1	21:07:32
ParentIdPrefix 8&a3d19e6&0		
Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102C0211&0	4/4	15:18:29
ParentIdPrefix 8&1d437750&0		
Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102D041B&0	4/1	21:07:32
ParentIdPrefix 8&6ae132f&0		
Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102104F1&0	4/4	15:18:16
ParentIdPrefix 8&2ffa93d7&0		
Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102B05EC&0	4/1	21:07:21
ParentIdPrefix 8&1d437750&0		

(they are all in ControlSet001, which according to System\Select\Current is the current set, ControlSet003 does not contain any extra devices either)

Since the device descriptors do not have a & as the second character, they are unique serial numbers (if the second character is a &, the OS created the descriptor for a device that didn’t have one – serial numbers are not required). Note that there is no reason for the descriptor to appear in the storage area of a USB memory stick that is capable of being imaged. A third party tool such as UVCView or USBView could be used with the USB device inserted into a PC or a hardware device such as Digital Intelligence's UltraBlock-USB could be used to observe the devices serial number.



A Tableau Forensic USB Bridge showing the hardware serial number for a USB disk device.

Also, since the devices all start with "Disk" and not CdRom, none of these are U3 devices.

HKLM\MountedDevices provides the driveletter assignments:
The ParentidPrefix value can be correlated with the Enum Devices entries listed above.

For example:
\DosDevices\G:
\??\STORAGE#RemovableMedia#**8&2ffa93d7&0**&RM#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}

So the USB Disk Memorex TD Classic 003B with a serial number 0778102104F1 was attached to the G drive letter.

The same process can be used for H:, I:, J: and K:

HKLM\SYSTEM\ControlSet001\control\DeviceClasses\53f5630d-b6bf-11d0-94f2-00a0c91efb8b
…USBSTOR\Disk&Ven_Flash&Prod_Drive_AU_USB20&Rev_8.07\BQENV5CR&0
…USBSTOR\Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102104F1&0
…USBSTOR\Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102B028C&0
…USBSTOR\Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102B05EC&0
…USBSTOR\Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102C0211&0
…USBSTOR\Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102C0441&0
…USBSTOR\Disk&Ven_Memorex&Prod_TD_Classic_003B&Rev_PMAP\0778102D041B&0
…USBSTOR\Disk&Ven_SAMSUNG&Prod_HD501LJ&Rev_0-10\152D203380B6&0
…USBSTOR\Disk&Ven_USB_2.0&Prod_Flash_Disk&Rev_1100\A127000000000063&0

These keys can provide last write times to the disks, but all times are right before the times listed above.

After performing all this manually, I found USBDeview, which automates obtaining this information on the local machine, a remote machine or an exported registry file (via the command line /regile option).

Much of the technique used above came from a paper by Victor Chileshe Lou which can be found here: http://scissec.scis.ecu.edu.au/conference_proceedings/2007/forensics/23_Luo_Tracing_USB_Device_artefacts_on_Windows_XP.pdf

Since none of this actually ties any of the USB images to the provided registry, I started looking into volume and file uniqueness. FDISK can report the volume id's of the USB images:

USB1	0xd2690c11 (volume id 0xf4d8dd43, name: TravelDrive, Fat16)
USB2	0xda6c3d8f (volume id 0xfd669053, name: TravelDrive, Fat16)
USB3	0x49d21600 (volume id 0x7c6c0ed5, name:NO NAME, Fat32)

(The bytes prior to the volume name are the volume id in little endian)

Unfortunately I have not yet been able to find references to the volume information in the registry.

The USB_3 image has "MSDOS5.0" in the Volume boot record which indicates that it was formatted by a Windows 2000 or XP machine.

Answers:

The follow devices have been attached to the system that the provided registry files came from.

Name	Description (Drive letter)	Vendor	Serial	Created date	ProductID
Drive AU_USB20	Flash Drive AU_USB20 USB Device	058f	BQENV5CR	3/12/2008 8:11:18 AM	6387
Flash Disk	USB 2.0 Flash Disk USB Device	090c	A127000000000063	2/14/2008 10:26:30 AM	1000
LaCie Hard Drive USB	SAMSUNG HD501LJ USB Device	059f	152D203380B6	2/28/2008 2:16:52 PM	0951
TD Classic 003B	Memorex TD Classic 003B USB Device(G):	12f7	0778102104F1	4/4/2008 8:18:03 AM	1d00
TD Classic 003B	Memorex TD Classic 003B USB Device(I:)	12f7	0778102B028C	4/1/2008 3:28:18 AM	1d00
TD Classic 003B	Memorex TD Classic 003B USB Device	12f7	0778102B05EC	3/21/2008 10:16:45 AM	1d00
TD Classic 003B	Memorex TD Classic 003B USB Device(H:)	12f7	0778102C0211	4/4/2008 8:18:08 AM	1d00
TD Classic 003B	Memorex TD Classic 003B USB Device(K:)	12f7	0778102C0441	4/1/2008 3:28:18 AM	1d00
TD Classic 003B	Memorex TD Classic 003B USB Device(J:)	12f7	0778102D041B	4/1/2008 3:28:18 AM	1d00

The three images provided in USB_1, USB_2 and USB_3 are likely Memorex TravelDrives based on the "ravelDrive" string present at the beginning of a partition (with the "e5" byte instead of the "T"), which means any of the images could be from one of the "TD Classic 003B devices." Similarly the Volume Boot Record for USB_1 and USB_2 both indicated FAT 16 "TravelDrive" drives.

Interestingly in the USB_002 image there are several recoverable and unrecoverable images. Among them are images found in other challenges. There are also references to the MSN yogibear1953 and Skype which are used in other challenges.

If a Memorex TravelDrive is found, the serial number can be inspected in order to link it to these registry files.

Tools used:
Type: Dedicated Forensics Tool
Name: Registry Viewer 1.3
Publisher: AccessData Corp
Commercial
Site: http://www.accessdata.com

Type: Inspection tool
Name: USBDeview
Publisher: Nirsoft
Freeware
Site: http://www.nirsoft.net/utils/usb_devices_view.html

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

204 Steganography Level 2

Methodology:

Steganography is a difficult problem space. We used several detection tools in an effort to limit the endless possibilities of "rabbit holes" that we might chase down. Essentially, we used publically available tools in order to get direction. In the end, Stegdetect while old and historically not particularly useful proved to be very useful in this particular case. Short execution remarks found below.

StegSpy 2.1
no results for any files.

XStegsecret
no results for any files.

StegalizerSS v3.1
File 1 shows some indication of have LSB encoded data.

Stegdetect 0.6
File 3 : appended (3250) <[random][data][7z..'….XL&U..]>

Stegbreak 0.6
no results for any files.

Answers:

File3 contains a 7zip archive starting at 0x2934E. This 7zip archive can be opened by any uncompressor that supports the 7zip format. The 7zip archive contains one file: mirage.bmp. This picture is of a several cars in a parking lot (one of them is a mirage) and an African American man getting into a red Mitsubishi Eclipse (or Eagle Talon) coupe. The image is in the recovered directory for challenge 204 on the media accompanying this report and can be seen below:

The 7zip file can be extracted using dd:

```
#>dd if=File3.jpg of=file3.7z bs=1 skip=168782
```

The 168782 is simply the starting offset (0x2934E) in decimal.

Tools used:
Type: specialized steganography detection software
Name: StegSpy
Publisher: Bill Englehardt
Compiled (free)
Site: http://www.spy-hunter.com/stegspydownload.htm

Type: specialized steganography detection software
Name: xstegsecret
Publisher: Alfonso Munoz
Open Source
Site: http://stegsecret.sourceforge.net/
Note: requires java (jre), and the GUI is in Spanish.

Type: specialized steganography detection software
Name: StegalizerSS
Publisher: SARC
Commercial (free trial – registration required)

Site: http://www.sarc-wv.com/stegalyzerss.aspx
Note: requires DotNet Framework and windows installer >=3.1

Type: specialized steganography detection software
Name: stegdetect
Publisher: Neils Provos
Open Source
Site: http://www.outguess.org/detection.php
Note: stegdetect is old. You likely need to set a shell variable CC=gcc34 (or gcc32 depending on your distribution) and have the compat-gcc-34 package installed.

Type: specialized steganography recovery software
Name: stegbreak
Publisher: Neils Provos
Open Source
Site: http://www.outguess.org/detection.php
Note: stegbreak 0.6 does not come with a rules.ini file. You need to create or procure one prior to running stegbreak. One option is to download stegbreak0.5 and use the rules.ini distributed with the older release.

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

301 – Encrypted Files and Folders

Methodology:

This challenge was not attempted

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

302 – Skype Communication Logs

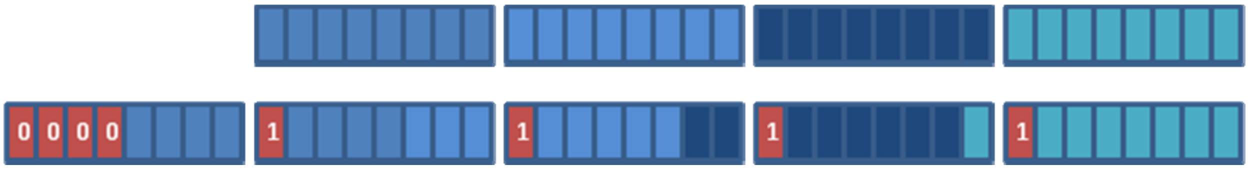
Methodology:

Even through the Unix strings utility provides much of the requested information, we decided to create some custom parsing software to generated human readable information from the data provided. Unfortunately at the time we wrote this software we did not have an internet connection and did not have java locally installed – so the solution was prototyped in perl and the written in C. Source code is provided on the media that accompanied this report.

Skype files are record based and the files are named according to function. For example a filename beginning with "chat" indicates chat logs, and would contain records for each event. The reasoning behind have chat256 and chat512 files is not known, though based on sample files it seems to be some sort of sorting technique based on maximum record size.

There are record headers (0x6c33336c) and identifying values for each member of the records. Several have been deciphered. Rather than enumerating these values here, see SkypeParser.h for details.

One part that I will elaborate on is the timestamp structure. Unix timestamps are stored in four bytes (seconds since Epoch). Skype timestamps are stored in five bytes. For unknown reasons the traditional four byte timestamp is expanded to 5 bytes by introducing four leading 0-bits, four bits of timestamp, 1 1-bit, 7 bits of timestamp, 1 1-bit, 7 bits of timestamp, 1 1-bit, 7 bits of timestamp, 1 1-bit, 7 bits of timestamp. One can only guess that this is supposed to obfuscate the true timestamp as there don't seem to be any performance or storage benefits for encoding the timestamp this way. In the graphic below the blue squares indicate bits in a Unix timestamp (the hue varies slightly by byte), the red squares indicate the additional bits present in a skype timestamp.



Almost all record types found contain a Sequence number, Timestamp and Username (among other type specific information). The provided software parses records found in call, chat, transfer, user profile, and voicemail files. Due to the unpublished nature of the records, the software makes "educated guesses" for ambiguous record data members.

One final note is that MessageID’s appear to be unique to the message and the same ID is used for both sides of the conversation. This means that if you have logs from both sides it can be shown that it is very likely that the two userids had communicated. The messageID and Sequence numbers can also be used to correlate multiple log files, while the provided software doesn’t currently do this, it would be fairly simple to add database functionality to facilitate easier search and display capabilities. Note that each execution of SkypeParser creates a SkypeParser.log file that contains comma separated values. This file can be used with Microsoft Excel, to populate a database, or a multitude or other applications.

Here is one sample execution:

```
#> ./skypeparser.exe chatmsg512.ddb
opening chatmsg512.dbb
The filename indicates a chat History

1 found record header at position 0
record size is 302 record ends at 136
record seq number is 32
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:00:18 2008
UserName  13:yogibear1953
Message  123:You know we still have that time thing going on, we miss our chance and we're out of luck
this time, maybe for a long time
DisplayName  15:Blane Stallman

2 found record header at position 208
record size is 292 record ends at 334
record seq number is 33
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:00:27 2008
UserName  9:kiki1932
Message  123:you know, we shouldn't be using names int htis converstion and yea I know about the time
thing but we gotta be careful man
DisplayName  9:Bob Zeus

3 found record header at position 410
record size is 309 record ends at 54d
record seq number is 38
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:02:00 2008
UserName  13:yogibear1953
Message  130:Yea, thought I was going to have a problem with the guns, by my uncle's a hunter and had
a lot so I just "borrowed" some from him
DisplayName  15:Blane Stallman

4 found record header at position 618
record size is 311 record ends at 757
record seq number is 40
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:02:50 2008
UserName  13:yogibear1953
Message  132:Naw, he keps them in the basement and hasn't used them in years.  they were in an old
metal cabinet, dusty and dirty as all get out
DisplayName  15:Blane Stallman
UserName  15:Ø⍰yogibear1953

5 found record header at position 820
record size is 303 record ends at 957
record seq number is 42
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:03:42 2008
UserName  13:yogibear1953
Message  124:Nope, used a rag over my nands and blew some dust back over where they had been
sitting. Things were a pain to clean though
DisplayName  15:Blane Stallman

6 found record header at position a28
record size is 299 record ends at b5b
record seq number is 48
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:05:11 2008
UserName  13:yogibear1953
Message  120:Got an old can of black powder from his basement also, maybe 30 pounds, old but never
been opened, should still be good
DisplayName  15:Blane Stallman

7 found record header at position c30
record size is 288 record ends at d58
record seq number is 50
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:05:47 2008
UserName  13:yogibear1953
Message  110:They would have wanted ID for that man, and I only have the one fake set and I didn't
want to burn it on that
DisplayName  15:Blane Stallman

8 found record header at position e38
record size is 290 record ends at f62
record seq number is 51
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:05:57 2008
UserName  9:kiki1932
Message  121:Good, whit what I got out of the anarchists cookbook combined with that were going to
open some eyesw I'll tell you that
```

```
DisplayName  9:Bob Zeus

9 found record header at position 1040
record size is 317 record ends at 1185
record seq number is 53
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:06:59 2008
UserName  9:kiki1932
Message  148:Except for some roving security and I told you I scoped that out and timed them up.
Always taking lunc together at the same time so we got an hour
DisplayName  9:Bob Zeus

10 found record header at position 1248
record size is 335 record ends at 139f
record seq number is 54
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:07:00 2008
UserName  13:yogibear1953
Message  156:I just don't want any mistakes.  It's one thing to do this but murder, man they sick a needle
in your arm and that stuff burning is the last thing you feel
DisplayName  15:Blane Stallman

11 found record header at position 1450
record size is 273 record ends at 1569
record seq number is 57
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:08:25 2008
UserName  9:kiki1932
Message  104:That was justt bad luck, and it was bad luck for them.  I didn't want anyone to get hurt,
you know that
DisplayName  9:Bob Zeus
MessageID  15:Ø⍰yogibear1953

12 found record header at position 1658
record size is 332 record ends at 17ac
record seq number is 59
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:09:07 2008
UserName  9:kiki1932
Message  163:Listen amigo you were right there too and unless you shut up and follwo the plan well both
be looking a a ride on the needle SO SHUT YOUR MOUTH ABOUT THE LAST JOB
DisplayName  9:Bob Zeus

13 found record header at position 1860
record size is 350 record ends at 19c6
record seq number is 60
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
```

```
TimeStamp  Sun Mar  2 06:09:43 2008
UserName  13:yogibear1953
Message  171:Don't talk to me that way.  I've been loyal and haven't said a thing.  I know they're still
looking for who pulled that and that means us.  I aint gonna help them kill me
DisplayName  15:Blane Stallman

14 found record header at position 1a68
record size is 380 record ends at 1bec
record seq number is 62
MessageID  41:#kiki1932/$yogibear1953;422ef3cb540f2cc2
TimeStamp  Sun Mar  2 06:10:39 2008
UserName  9:kiki1932
Message  211:listen, we got this going and just need to chill awhile.  Gotta get off this comm and get on
the other one we set up so they cant trace us so good.  Get up on that one, regular time and well finish
the planning
DisplayName  9:Bob Zeus
```

Answers:

See the above sample run. Similar results can be found for each file (and correlated to each other). See the 302 solution directory on the media that accompanies this report.

Tools used:

None other than java. Common tools like xxd were used somewhat during the development process, but not during analysis.

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

303 – Language Identification

Methodology:

The general concept is to provide a software mechanism that “understands” the Word .doc format enough to distinguish text from metadata and that can detect the language of the text from the text itself (and optionally translate it).

Initially a PERL prototype was created, with the idea of using metadata to obtain language information. For example each document has an MSOEncoding field which contains values such as 1256 for Arabic or 949 for Korean (see meta.pl for a large listing of these codes). This approach has several draw backs: some encodings are very broad, such as 65000 for “UTF.” UTF doesn’t imply much about the language used. The use of the MSOEncoding field (and other fields) cannot be guaranteed: do different versions of office use different encodings (for the same text one version may use the specific language code, while another uses a more general code). Does the installation, presence, or selection of a Windows Language Pack effect Office in a way that alters some or all of these fields? Finally, when using PERL’s Win32::OLE module, the system that uses the module (the one executing meta.pl) must actually have Microsoft Word installed because the module actually uses the Word application to perform tasks. In fact you can observe the tasks by setting the Win32::OLE object to visible:

```
    my $wordobj = Win32::OLE -> new(‘Word.Application’, ‘Quit’) or die “Word app not installed”;
    $wordobj -> {visible} = 1;  #allows you to see the window
```

This reliance on the Microsoft Word application has several disadvantages including the requirement to obtain a license for the software, the possibility of introducing your own metadata into the document (revision count, history etc), and versioning issues. Depending on the Version of Word installed, the document could conceivable be interpreted differently which makes results difficult to reproduce across heterogeneous machines.

The second prototype was also done in PERL, but this focused on the text instead of metadata. A few language modules exist such as Lingua::Translate and WWW::Babelfish. Modules such as these contain various methods of detection and translation. The file 303.pl demonstrates the possibility of using Win32::OLE to extract text from Word doc files then using the language modules do detect / translate the text. This still requires Microsoft Word to be installed and suffers from that reason suffers from many of the same issues mentioned above. A more robust solution was needed.

To alleviate the Microsoft Word Application requirement, a parser capable of parsing the entire word file independent of installed Applications was needed. The POI project by the Apache Software Foundation fulfills this need. POI (poi.apache.org) is a pure java implementation of several OLE2 Microsoft file formats (and others). In particular, for this challenge the POI HWPF (Horrible Word Processor Format) functions were used. In addition to allowing easy access to the text portions of the doc files, POI also facilitates access to much of the metadata that we observed through meta.pl. Accessing the text portions of the files could be done several ways, in the end we settled on obtaining text from a the doc one paragraph at a time – this choice was mainly made to support the language detection/translation efforts.

For the language features, after experimenting with several translation techniques and services we chose Google's translate service. There are several reasons to use this service rather than attempt to create translation software: A) we are not linguists, Google employs linguists and translators, B) Google's translation service will continue to improve after the DC3 challenge is over essentially making our solution to this challenge better over time, C) Google translate supports many languages and language pairs, and D) a java implementation to the Google Translate API already existed.

Rich Midwinter created google-api-translate-java (available on code.google.com). After researching the Google translate API, we found that while the java implementation was a good start (and in practice worked very well) some features of the API were not readily usable using the java package. So we added some functionality to the java translate package – in particular we added automatic language detection (The source for the patch from 4.0 to 4.0a, and scripts to rebuild the jar file are included on the media that accompanies this report).

Since the Google translate service enforces a maximum character limit, and the URL Encoding of text may result in a very large string (especially in some languages like Russian), the translation service can not translate the entire text at one time. POI offers the granularity of a paragraph, but even then some paragraphs are too large once encoded. One approach is to simply cut the paragraph into sections that are small enough for translation. This approach could split sentences into two parts which may affect the translation accuracy. Instead we use java's BreakIterator class which can be instantiated with a specific locale. This way we can perform operations at the *sentence* granulariy. Once the language is detected the appropriate locale can be used. Since many languages used different punctuation rules (like not ending sentences with a period) use of the BreakIterator class with the correct locale is very beneficial.

The basic algorithm the software follows is:

1. Create a POI object for each .doc file that is specified
2. Do language detection tests at random points in the file across large ranges of text
3. Get the text from the document one paragraph at a time
4. Perform translation one sentence at a time

For challenge 303, the software does not need to perform steps 3 and 4.

Various choices made when creating the software allow for other interesting usage scenarios – for example one doc file that contains several languages, like one paragraph in English, one in French and one in Spanish (and MultiLanguage.doc file is included to demonstrate this capability).

Currently the software solution for challenge 303 requires google-api-translate-java-0.4.jar, json.jar, poi-3.0.2 (scratchpad, FINAL, and contrib. jars) and the software written for this challenge.

To build everything execute:

```
#>./build_gexample_and_translator.sh
```

Once built, to detect (or translate) two Word documents (Document1Tx.doc and Document2Tx.doc):

```
#> java -cp .:./google-api-translate-java-0.4a.jar:./poi-3.0.2-
FINAL/poi-scratchpad-3.0.2-FINAL-20080204.jar:./poi-3.0.2-FINAL/poi-
3.0.2-FINAL-20080204.jar:./poi-3.0.2-FINAL/poi-contrib-3.0.2-FINAL-
20080204.jar c303 Document1Tx.doc Document2Tx.doc
```

Note that from “java” to “Document2Tx.doc” is one line. You can make this smaller by placing the jar files in your classpath instead of specifying the classpath on the command line. The locations to place jar files and syntax for specifying your classpath vary by operating system.

```
#> java c303 Document1Tx.doc Document2Tx.doc
```

Omitting the .doc arguments results in the display of a usage statement. The program can handle 0 or more .doc files specified on the command line. And the translation language can be forced (be default it automatically detects the language to translate from – it always translates to English).

A sample build and a sample run are available in an appendix.

Answers:

File Name	Original Language	Language Code
Document 1Tx.doc	Chinese Traditional	zh-TW
Document 2Tx.doc	Russian	ru
Document 3Tx.doc	Korean	ko
Document 4Tx.doc	Arabic	ar
Document 5Tx.doc	Japanese	ja

Initial indications:
Document2Tx.doc – Russian, large use of Cyrillic (0x04xx) characters
Document3Tx.doc – Korean, large use of Hangul script elements
Document4Tx.doc – Arabic, large use of 0x06xx characters

Final answers were obtained by use of the java tool developed for this challenge.

Tools used:
Type: java interface to the Google Translate service
Name: Google Translate API JAVA
Publisher: Rich Midwinter
Open Source
Site: http://code.google.com/p/google-api-translate-java/
Note: For use in this challenge this package was modified. The patch and a patched version are on the media that accompanies this report.

Type: Java Object container
Name: JSON
Publisher:
Commercial / Open Source
Site: http://www.json.org/
Note: download from json.org. Uncompress (preserve directory structure). Compile with "javac *.java" then go back to the root directory and create a jar: "jar –cvf json.jar org\json\*.class" then the json.jar can be placed in the same directory or somewhere in your java path.

Type: OLE java interface
Name: POI
Publisher: Apache Software Foundation
Open Source
Site: http://poi.apache.org

Type: perl module
Name: Win32::OLE
Publisher: Gurusamy Sarathy
Open Source
Site: http://search.cpan.org/~jdb/Win32-OLE-0.1709/lib/Win32/OLE.pm

Type: perl module
Name: Lingua::Translate
Publisher: Sam Vilain
Open Source
Site: http://search.cpan.org/~samv/Lingua-Translate-0.09/lib/Lingua/Translate.pm

Type: perl module
Name: WWW::Babelfish
Publisher: Dan Urist
Open Source
Site: http://search.cpan.org/~durist/WWW-Babelfish-0.16/Babelfish.pm

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

401 – MSN Session Logs and Chat

Methodology:

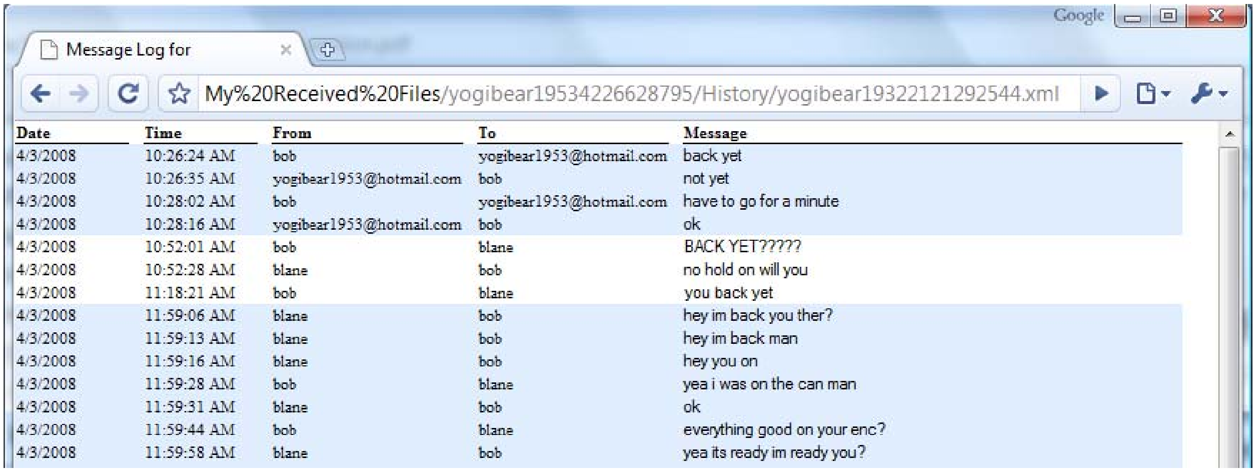
Initially we sought to take an approach similar to challenge 302. But cursory inspection of the MSN files revealed that the information we wanted was already in XML form! Typically found in:

C:\Documents and Settings\<windows login>\My Documents\My Received Files\<MSN login>\History

is a file that starts with the left-hand-side of the email address of the person <MSN login> was chatting with followed by several numbers and .xml. For example if you are chatting with robert@microsoft.com the filename might be robert12345678.xml. Inside this file you would find something like:

Date	Time	From	To	Message
1.1.2008	12:01:01	Person1	Person2	"Message text here"

There is even a style sheet (MessageLog.xsl) that makes the xml display nicely in a web browser. One part that is not readily apparent in the browser display is that the xml file also contains incrementing session identifiers (they are indicated by alternating white and blue highlighting.) The XML files are created if the "automatically keep a history of my conversations" box is checked.


Google
Message Log for
My%20Received%20Files/yogibear19534226628795/History/yogibear19322121292544.xml

Date	Time	From	To	Message
4/3/2008	10:26:24 AM	bob	yogibear1953@hotmail.com	back yet
4/3/2008	10:26:35 AM	yogibear1953@hotmail.com	bob	not yet
4/3/2008	10:28:02 AM	bob	yogibear1953@hotmail.com	have to go for a minute
4/3/2008	10:28:16 AM	yogibear1953@hotmail.com	bob	ok
4/3/2008	10:52:01 AM	bob	blane	BACK YET?????
4/3/2008	10:52:28 AM	blane	bob	no hold on will you
4/3/2008	11:18:21 AM	bob	blane	you back yet
4/3/2008	11:59:06 AM	blane	bob	hey im back you ther?
4/3/2008	11:59:13 AM	blane	bob	hey im back man
4/3/2008	11:59:16 AM	blane	bob	hey you on
4/3/2008	11:59:28 AM	bob	blane	yea i was on the can man
4/3/2008	11:59:31 AM	blane	bob	ok
4/3/2008	11:59:44 AM	bob	blane	everything good on your enc?
4/3/2008	11:59:58 AM	blane	bob	yea its ready im ready you?

Example of viewing yogibear19322121292544.xml in the Chrome web browser.

I'm not certain how yogibear1953\History\yogibear1953 occurred. In my testing I couldn't re-create a scenario where msn created a history with the same account name as the login account. If you log in with a different account the history file is saved under the directory tree for that new account.

We looked into the .sqm (sqmdata) files for some time, but we are still not convinced that they hold any additional information. If SQM (Software Quality Management) or CEIP (Customer Experience Improvement Program) are enabled, then additional data collection information may be available such as: sign in statistics, message delivery success rates, disconnection percentages, audio/video failures, and possibly information about the local hardware and/or software installed. The exact types of information tracked and the file format used for storage is not publically available. These files are created if the "allow Microsoft to collect information" box is checked.

The software written for this challenge parses the xml, and displays information not readily observable in a web browser (such as the session id information). Additionally the software attempts to detect display name changes (note how yogibear1953 switch to blane is automatically detected) and has the ability to produce PostgreSQL compatible output for use with a database. Once the information is in the database it can be sorted and viewed in many difference ways.

Answers:

A full run of the java tool can be found in an appendix, this includes to/from, message and timestamp information as requested (about 6 pages). Sourcecode, sample runs, and SQL output can be found on the media that accompanies this report.

The three files are from Blaine's computer and they indicate he yogibear1953@hotmail (blaine or blane) is also yogibear1932@hotmail and he chatted with udorntani@_____ (zues / bob).

1932.xml indicates that bob=yogibear1932	("from" in addition to the filename)
yogibear1953@hotmail = blaine	(via a changed display name)
1953.xml indicates that yogibear1953 = blane	("to" plus the filename)
zues = bob	(via changed displayname)
udorntani.xml indicates that udorntani = zues = bob	(loosely based on chat text; manual inference)

Furthermore, Documents and Settings\<windows name>\Contacts\<msn login> indicates that "Tall Tower" is the computer login name for yogibear1953 (blane).

\MSN 2\MSN Messenger\Contacts for yogibear1953 (hotmail).cct has one contact in it: yogibear1932@hotmail (MSN Messenger can be used to open the cct file). Since the filename is "yogibear1953" the MSN 2 directory likely came from Blane's computer – though that should be verified with whomever collected the evidence.

And the same is true for MSN 1. In fact md5sum and diff for files in MSN 1 and MSN 2 show that the directories are identical except that MSN 1 contains these four files in /MSN 1/MSN Messenger/My Received Files/ yogibear19534226628795/History/ (while MSN 2 does not):

MessageLog.xsl
udorntani1744684863.xml
yogibear19322121292544.xml

yogibear19534226628795.xml

Note that the 1953 part of yogibear1953 is part of the username. The remaining 10 digit numbers appear to be a unique MSN ID assigned to the user though that is only supposition (similar for the other files).

Tools used:
Type: Messaging utilitiy
Name: MSN Messenger
Publisher: Microsoft Corp
Commercial
Site: http://webmessenger.msn.com (messenger come preinstalled on many XP and Vista systems)

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

402 Image Analysis

Methodology:

Since the images are the same ones used in 103, preliminary analysis can be done using the same approach.

Answers:

The files' metadata was inspected for presence of information indicating a 'real' picture (such as camera information), and for information indicating 'not real' picture (such as known editing tools). The process is similar to that for challenge 103. The metadata output can be found on the media that accompanies this report.

Girl_with_glasses is likely fake or altered by the GD image library. While the library's original intent may have been to create images, there are now many uses of the library in many programming languages (such as thumbnailing, cropping, resizing, etc of existing images.)

	Real	enhanced	CG	Composite	Notes
beach_foot.jpg	yes	possibly	no	no	Camera metadata, photoshop
bearded_guy.jpg	possibly	possibly	no	no	looks real.
blue_eyes.jpg.jpg	no	yes	no	yes	photoshop info, appears edited
board.jpg	yes	no	no	no	no indication of editing
bride.jpg	yes	yes	no	no	photoshop,
brunette.jpg	yes	no	no	no	camera metadata
curtain_lady.jpg	partial	no	poss.	Yes	curtains appear cg, photoshop
eagle.jpg	yes	no	no	no	camera metadata

elder_lady.jpg	partial	no	no	yes	visible watermark
girl.jpg.jpg	possibly	possibly	poss.	No	appears possibly CG
girl_with_glasses.jpg	no	no	yes	no	appears CG
guy.jpg	yes	possibly	no	no	appears real
keyboard.jpg	yes	no	no	no	camera metadata
lady_in_tshirt.jpg.jpeg	possibly	possibly	poss.	Possibly	picture is too small to tell.
leonardo.jpg	yes	yes	no	no	from film
pensive.png	yes	no	no	no	appears real
sequin_girl.jpg	yes	yes	no	no	appears altered, very small.
spoons.bmp	yes	no	no	no	appears real.
tulips.jpg	yes	no	no	no	camera metadata
veiled_lady.jpg	no	no	yes	no	appears cg

Tools used:
Type: meta data extrator
Name: exiftool
Publisher: Phil Harvey
Open Source (Perl License)
Site: http://www.sno.phy.queensu.ca/~phil/exiftool

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

403 – Language Text Translation

Methodology:

Much of the methodology described in 303 applies to 403 as well. Actually the same java program is submitted for both challenges. The five provided files were the same for both challenges. Translated text can be found below.

Larger sample sets and large amounts of original and translated documents for comparison would make the translation capabilities better. Using POI the software could easily put translated text back into .doc documents, since this was not required this capability has not yet been added. This capability might be desirable due to inclusion of embedded objects like photographs.

Answers:

The following translations were not edited or enhanced by a human in any way. These are translations taken direction from the tool provided.

File Name	Original Language	Text After Translation to English
Document 1Tx.doc	Chinese Traditional	EN_Document1Tx.txt
Document 2Tx.doc	Russian	EN_Document2Tx.txt
Document 3Tx.doc	Korean	EN_Document3Tx.txt
Document 4Tx.doc	Arabic	EN_Document4Tx.txt
Document 5Tx.doc	Japanese	EN_Document5Tx.txt

Document1Tx.doc
Language: chinese (trad)
I wrote my paper in the seminar of the “three strikes and you are in the” model, this is all over the history of the United States, it is applied to it reached the Virginia Federation, and was aware of some of the successes and failures. I also estimated that the views of those in Virginia and outside the criminal law system, and draw a conclusion on what the future may bring As for the model and its further use in the state. Internet will be the place where I started my research. By using a different search engine, I found the article had been written on the three strikes policy in the country and their application in the vicinity. The articles from the news media, Crime and Justice Foundation think-tank, and legal information website. These articles help to draw pictures of what to do in the criminal law system on the three-strike style, and they can provide the perspective that it impact on the policy.

Document2Tx.doc
Language: russian
My next step will be to build uchreditelstvo my paper, through the study and discuss the history of 3 strikes policy, raised its enactment of 23 provisions of the connection, the main difference is between those provisions and details of policies enacted within the power of the legislature of Virginia. One of the key parts of the legislation, which I will discuss the management act and law enforcement 1994 violent crime and politics that followed around the country. For more information, I will use the database provided that the archive of the University of Norwich, and if necessary, direct contact with the Ministry of Justice Virginia reached Database that can provide information accessible databases of the university identified. My following the step will cover the perceived successes and failures of models in its 14 year history. To do this, I will use the reviews of specific business situations to illustrate how policies are made to individual criminals, and the results of the application were. To find this information, I will continue to use the database schools, and the information that I can glean through the materials collected from the Ministry of Justice and the Virginia corrections department.

Document3Tx.doc
Language: korean
My next step, some of the leading experts in criminal law system in the country 3 1 week it hit the application of a model talking about, some of the interest groups in the future, where legislation to try to map what to show and explain To withdraw the nation&#39;s great to be back in view. I am of the Commonwealth of Virginia must continue to use it to that statement of how I feel, or it presents a possible alternative to the use of a Dodge models hit 3 of the development of the past 14 years is how to get out through the papers In fact, to discuss with the conclusion of my paper, the discussion will be, and complete conclusion. Through this paper, the statement I will be a deeper understanding of the relationship between the federal crime of national population policies and their different effects on the development of crime and the show. Approximately 400 people convicted in Virginia of proven 3 batting villains (3 hours loser) is the time serving under the law. The regulations in the early 1980s. I swear to the law for anyone convicted deleted was a strong violation of the separation of about 3. For the previous 2 hours by individuals for such crimes to crimes against the service after being proven guilty, Virginia to life in prison for the prosecution eoteulgi they can now run, so that was a low probability.

Document4Tx.doc
Language: arabic
This annual conference, co-sponsored by the Ministry of Defense and the Center for Online Crimes Task Force - Global Network Operations, The leaders of government and industry together to discuss all aspects of computer crime: entering the investigation of cyber crime law, digital forensics, information assurance, as well as Research and development, testing and evaluation of digital forensic tools. The goal of this year to attend the conference was to share knowledge and processes as well as to address current and emerging threats. “This is all a bit of cyber crime conference, and not just the latest information assurance,” said Special Agent (retired) Jim Christy, director of futures exploration at the center of the Ministry of Defense in cyber crime. “Offers that change every year to address the current trends and tools. “We will have hands on the digital forensic training available in two sessions for two days this year. Last year, more than 240 of the 700 attendees took advantage of our hands - on training. Or that the demand

Document5Tx.doc
Language: japanese
Law enforcement from around the world, anti-information, and information assurance experts from the threat of cyberspace, the growth of our partners to focus on the protection and training to meet for discussions. Cyber crime in the 7th annual meeting in January 2008 Department of Defense 14-18 St. Louis, Missouri in the Renaissance Retain is a magnificent hotel.   January 2008 training will be holding 11-14. The main speaker, the Eighth Air Force commander Lt Gen. : Robert J. “Bob” Elder, Jr. The Air Force&#39;s new Cyber to discuss the establishment of order.   The meeting of cyber crime prevention training and execution of many of the features offered by a leader. Conference and exhibition company that features related to the agency. All meetings of the central government, the state is open, and the following work in the field of defense industry infrastructure and civilians from the building contractor of DoD, as well as local law enforcement,: criminal investigator, special agent-to-information

Tools used:
See challenge 303.

Team Name: f0gd0gs

Results Email: [redacted]

Examination Time Frame: April 14, 2008 to Oct 29, 2008

404 – Steganography – Level 4

Methodology:

We have nothing to report on this challenge.

Appendix 1 Dumpreg.pl

```
#!/usr/bin/perl

# http://search.cpan.org/src/JMACFARLA/Parse-Win32Registry-0.23/bin/dumpreg.pl
# Note: you must first install the library:
#~>cpan
#cpan>install Parse::Win32Registry
#cpan>exit
```

```
use strict;
use warnings;

binmode(STDOUT, ":utf8");

use Parse::Win32Registry;
use Getopt::Long;

Getopt::Long::Configure('bundling');
Getopt::Long::Configure('prefix_pattern=(--|-|\+|\/)');
Getopt::Long::Configure('long_prefix_pattern=(--|\/)');

my $debug;
my $quiet;
my $recurse;
my $indent = 0;
GetOptions('debug|d'   => \$debug,
           'quiet|q'   => \$quiet,
           'recurse|r' => \$recurse,
           'indent|i'  => \$indent);

my $filename = shift or die usage();
my $initial_key_name = shift;

my $registry = Parse::Win32Registry->new($filename);
my $root_key = $registry->get_root_key;

if (defined($initial_key_name)) {
    $root_key = $root_key->get_subkey($initial_key_name);
    if (!defined($root_key)) {
        die "Could not locate '$initial_key_name' key in '$filename'\n";
    }
}

traverse($root_key);

sub traverse {
    my $key = shift;
    my $pathname = shift || "";
    my $depth = shift || 0;

    if ($indent) {
        print "  " x ($depth * $indent);
    }
    else {
        print "\n" if !$quiet;
        print "$pathname";
    }
        $debug ? $key->print_debug : $key->print_summary;

    if (!$recurse) {
        foreach my $subkey ($key->get_list_of_subkeys) {
            print "  " x ($depth * $indent);
            print "= ", $subkey->get_name, " (key)\n";
        }
    }

    if (!$quiet) {
        foreach my $value ($key->get_list_of_values) {
            print "  " x ($depth * $indent);
            print "- ";
            $debug ? $value->print_debug : $value->print_summary;
        }
    }

    $pathname .= $key->get_name . "\\";
    if ($recurse) {
        foreach my $subkey ($key->get_list_of_subkeys) {
            traverse($subkey, $pathname, $depth + 1);
        }
    }
}

sub usage {
    return <<USAGE;
dumpreg.pl for Parse::Win32Registry $Parse::Win32Registry::VERSION

dumpreg.pl <filename> [subkey] [-r] [-q] [-i] [-d]
    -r or --recurse     traverse all child keys from the root key
                        or the subkey specified
    -q or --quiet       do not display values
    -i or --indent      indent subkeys and values to reflect their
                        level in the registry tree
    -d or --debug       display debugging information about
                        subkeys and values
USAGE
}
```

Appendix 2 Sample build of challenge 303 (and 403) c303.java

```
# ./build_gexample_and_translatejar.sh
added manifest
adding: com/(in = 0) (out= 0)(stored 0%)
adding: com/google/(in = 0) (out= 0)(stored 0%)
adding: com/google/api/(in = 0) (out= 0)(stored 0%)
adding: com/google/api/translate/(in = 0) (out= 0)(stored 0%)
adding: com/google/api/translate/Language$LanguageDesc.class(in = 529) (out= 327)(deflated 38%)
adding: com/google/api/translate/Language.class(in = 5043) (out= 2297)(deflated 54%)
adding: com/google/api/translate/Translate.class(in = 6161) (out= 2883)(deflated 53%)
ignoring entry META-INF/
ignoring entry META-INF/MANIFEST.MF
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/json/(in = 0) (out= 0)(stored 0%)
adding: org/json/JSONObject.class(in = 16145) (out= 7628)(deflated 52%)
adding: org/json/JSONArray.class(in = 10698) (out= 4631)(deflated 56%)
adding: org/json/JSONException.class(in = 751) (out= 434)(deflated 42%)
adding: org/json/JSONString.class(in = 156) (out= 121)(deflated 22%)
adding: org/json/JSONTokener.class(in = 5210) (out= 2893)(deflated 44%)
adding: org/json/JSONObject$Null.class(in = 768) (out= 400)(deflated 47%)
Detected Language: french
Translated Text  : This is a sample of a translation program written in Java
Note: c303.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

```
# java -cp .:./google-api-translate-java-0.4a.jar:./poi-3.0.2-FINAL/poi-scratchpad-3.0.2-FINAL-20080204.jar:./poi-3.0.2-FINAL/poi-3.0.2-FINAL-20080204.jar:./poi-3.0.2-FINAL/poi-contrib-3.0.2-FINAL-20080204.jar c303

this program translates word documents to english using google translate

the program requires 1 or more file names as arguments

Options:
 -f <lang>  forces translation to a chosen language
         must be followed immediately by an ISO 639 language code.  (like en,it,fr,ru...)
 -v        increases verbosity (may be used multiple times)
 -s        'simple' mode.  Only displays the translated text, no extra information
```

Appendix 3 Sample execution of challenge 303 (and 403)

```
# java -cp .:./google-api-translate-java-0.4a.jar:./poi-3.0.2-FINAL/poi-scratchpad-3.0.2-FINAL-
20080204.jar:./poi-3.0.2-FINAL/poi-3.0.2-FINAL-20080204.jar:./poi-3.0.2-FINAL/poi-contrib-3.0.2-FINAL-
20080204.jar c303 Document*Tx.doc

Detected Language: chinese (trad)(zh-TW - will translate as zh)
Translated Text  : âI wrote my paper in the seminar of the &quot;three strikes and you are in the&quot;
model, this is all over the history of the United States, it is applied to it reached the Virginia Federation,
and was aware of some of the successes and failures.
Translated Text  : I also estimated that the views of those in Virginia and outside the criminal law
system, and draw a conclusion on what the future may bring As for the model and its further use in the
state.
Translated Text  : Internet will be the place where I started my research.
Translated Text  : By using a different search engine, I found the article had been written on the three
strikes policy in the country and their application in the vicinity.
Translated Text  : The articles from the news media, Crime and Justice Foundation think-tank, and legal
information website.
Translated Text  : These articles help to draw pictures of what to do in the criminal law system on the
three-strike style, and they can provide the perspective that it impact on the policy.

Detected Language: russian(ru)
Translated Text  : My next step will be to build uchreditelstvo my paper, through the study and discuss
the history of 3 strikes policy, raised its enactment of 23 provisions of the connection, the main
difference is between those provisions and details of policies enacted within the power of the legislature
of Virginia.
Translated Text  : One of the key parts of the legislation, which I will discuss the management act and
law enforcement 1994 violent crime and politics that followed around the country.
Translated Text  : For more information, I will use the database provided that the archive of the
University of Norwich, and if necessary, direct contact with the Ministry of Justice Virginia reached
Database that can provide information accessible databases of the university identified.
Translated Text  : My following the step will cover the perceived successes and failures of models in its
14 year history.
Translated Text  : To do this, I will use the reviews of specific business situations to illustrate how
policies are made to individual criminals, and the results of the application were.
Translated Text  : To find this information, I will continue to use the database schools, and the
information that I can glean through the materials collected from the Ministry of Justice and the Virginia
corrections department.

Detected Language: korean(ko)
Translated Text  : My next step, some of the leading experts in criminal law system in the country 3 1
week it hit the application of a model talking about, some of the interest groups in the future, where
legislation to try to map what to show and explain To withdraw the nation&#39;s great to be back in
view.
Translated Text  : I am of the Commonwealth of Virginia must continue to use it to that statement of
how I feel, or it presents a possible alternative to the use of a Dodge models hit 3 of the development of
```

the past 14 years is how to get out through the papers In fact, to discuss with the conclusion of my paper, the discussion will be, and complete conclusion.
Translated Text : Through this paper, the statement I will be a deeper understanding of the relationship between the federal crime of national population policies and their different effects on the development of crime and the show.
Translated Text : Approximately 400 people convicted in Virginia of proven 3 batting villains (3 hours loser) is the time serving under the law. The regulations in the early 1980s.
Translated Text : I swear to the law for anyone convicted deleted was a strong violation of the separation of about 3.
Translated Text : For the previous 2 hours by individuals for such crimes to crimes against the service after being proven guilty, Virginia to life in prison for the prosecution eoteulgi they can now run, so that was a low probability.

Detected Language: arabic(ar)
Translated Text : This annual conference, co-sponsored by the Ministry of Defense and the Center for Online Crimes Task Force - Global Network Operations, The leaders of government and industry together to discuss all aspects of computer crime: entering the investigation of cyber crime law, digital forensics, information assurance, as well as Research and development, testing and evaluation of digital forensic tools.
Translated Text : The goal of this year to attend the conference was to share knowledge and processes as well as to address current and emerging threats.
Translated Text : <br> &quot;This is all a bit of cyber crime conference, and not just the latest information assurance,&quot; said Special Agent (retired) Jim Christy, director of futures exploration at the center of the Ministry of Defense in cyber crime.
Translated Text : &quot;Offers that change every year to address the current trends and tools.
Translated Text : <br> &quot;We will have hands on the digital forensic training available in two sessions for two days this year.
Translated Text : Last year, more than 240 of the 700 attendees took advantage of our hands - on training.
Translated Text : Or that the demand

Detected Language: japanese(ja)
Translated Text : Law enforcement from around the world, anti-information, and information assurance experts from the threat of cyberspace, the growth of our partners to focus on the protection and training to meet for discussions.
Translated Text : Cyber crime in the 7th annual meeting in January 2008 Department of Defense 14-18 St. Louis, Missouri in the Renaissance Retain is a magnificent hotel.
Translated Text : January 2008 training will be holding 11-14.
Translated Text : The main speaker, the Eighth Air Force commander Lt Gen.
Translated Text : Robert J. &quot;
Translated Text : Bob &quot;Elder, Jr.
Translated Text : The Air Force&#39;s new Cyber to discuss the establishment of order.
Translated Text : The meeting of cyber crime prevention training and execution of many of the features offered by a leader.
Translated Text : Conference and exhibition company that features related to the agency.

Translated Text : All meetings of the central government, the state is open, and the following work in the field of defense industry infrastructure and civilians from the building contractor of DoD, as well as local law enforcement,: criminal investigator, special agent-to-information

Appendix 4 Word Document Translation and Detection program source code

```
//Tim Vidas DC3 2008 Challenge, 303 and 403

import java.io.*;
//import java.net.*;
//import java.sql.*;
import java.util.*;
import org.apache.poi.hwpf.*;
import org.apache.poi.*;
import org.apache.poi.hwpf.extractor.*;
import org.apache.poi.POITextExtractor.*;
import com.google.api.translate.Language;
import com.google.api.translate.Translate;
import java.text.BreakIterator;
import java.net.URL;  //only needed for using urlEncodeText

import org.json.JSONObject;

//http://icebox.uc.nps.edu/google-translate-docs/
//http://java.sun.com/j2se/1.4.2/docs/api/java/text/BreakIterator.html
//http://www.google.com/support/talkgadget/bin/answer.py?answer=89921#
//http://code.google.com/apis/ajaxlanguage/documentation/reference.html#ResultObjects
//http://mirror.candidhosting.com/pub/apache/poi/release/bin/
//http://code.google.com/p/google-api-translate-java/
//http://java.sun.com/j2se/1.4.2/docs/api/java/util/Locale.html
//http://forums.asp.net/p/1218210/2171728.aspx#2171728
//http://www.json.org/javadoc/org/json/JSONObject.html
//http://java.sun.com/j2se/1.4.2/docs/api/java/net/URLConnection.html#getInputStream()


public class c303 {
   public static final int TRANSLATOR_LIMIT = 800;
   private static int VERBOSITY = 1;
   private static final int V_INFO = 1;
   private static final int V_BASIC = 2;
   private static final int V_META = 3;
   private static final int V_EXTRA = 4;
   private static final int V_DEBUG = 30;


   public static LinkedList<URL> getChunksDumb (String[] paragraphs){
      LinkedList<URL> returnblocks = new LinkedList();
      try {
        for ( int i = 0; i < paragraphs.length; i++ ) {
              Sout("Paragraph  " + i + ": " + paragraphs[i].length() + " chars",
V_BASIC);
              Sout("UrlEncoded " + i + ": " +
((Translate.urlEncodeText(paragraphs[i],"auto","english")).toString()).length() + "
chars", V_BASIC);
           //don't want to break in the middle of a character, so keep encoding until
you get the right size - ideally this would be on a 'sentence boundary'
           while ( paragraphs[i].length() > 0 ) {
              URL potentialBlock =
Translate.urlEncodeText(paragraphs[i],"auto","english");
              int j = paragraphs[i].length();
              //Sout("\npara size is " + paragraphs[i].length() + " j is " + j);
              while ( potentialBlock.toString().length() > TRANSLATOR_LIMIT ) {
                j = j - 20;  //very crude, assumes translator can handle 20 and make
break up words, sentences, etc
```

```
                potentialBlock =
Translate.urlEncodeText(paragraphs[i].substring(0,j),"auto","english");
              }
              returnblocks.addLast(potentialBlock);
              //          Sout(" Block " + returnblocks.size() + ": " +
potentialBlock.toString().length());
              paragraphs[i] = paragraphs[i].substring(j,paragraphs[i].length());
//cut out the already chunked block
              //Sout("para size is " + paragraphs[i].length() + " j is " + j);
              if( returnblocks.size() > 20) {
                break;
              }
           }
        }
      } catch (Exception ex) {
        ex.printStackTrace();
      }
      return returnblocks;
   }

   public static LinkedList<URL> getChunksBySentence (String[] paragraphs, Locale
loc){
      LinkedList<URL> returnblocks = new LinkedList();
      try {

        //new Locale("en","US")
        String lang = loc.getLanguage();
        BreakIterator sentenceIterator = BreakIterator.getSentenceInstance(loc);


        for ( int i = 0; i < paragraphs.length; i++ ) {
              Sout("Paragraph  " + i + ": " + paragraphs[i].length() + " chars",
V_BASIC);
              Sout("UrlEncoded " + i + ": " +
((Translate.urlEncodeText(paragraphs[i],lang,"english")).toString()).length() + "
chars", V_BASIC);
           //don't want to break in the middle of a character, so keep encoding until
you get the right size - ideally this would be on a 'sentence boundary'
           //   while ( paragraphs[i].length() > 0 ) {
           //      int j = paragraphs[i].length();
           //Sout("\npara size is " + paragraphs[i].length() + " j is " + j);

           sentenceIterator.setText(paragraphs[i]);
           int start = sentenceIterator.first();
           int end = sentenceIterator.next();

           while (end != BreakIterator.DONE) {
              String sentence = paragraphs[i].substring(start,end);
              URL potentialBlock = Translate.urlEncodeText(sentence,"auto","english");
              returnblocks.addLast(potentialBlock);
              //              Sout(" Block " + returnblocks.size() + ": " +
potentialBlock.toString().length());
              //Sout(sentence);
              start = end;
              end = sentenceIterator.next();
           }

           //      while ( potentialBlock.toString().length() > TRANSLATOR_LIMIT ) {
           //        j = j - 20;  //very crude, assumes translator can handle 20 and
make break up words, sentences, etc
           //        potentialBlock =
Translate.urlEncodeText(paragraphs[i].substring(0,j),"auto","english");
```

```
           //       }
           //       paragraphs[i] = paragraphs[i].substring(j,paragraphs[i].length());
//cut out the already chunked block
           //Sout("para size is " + paragraphs[i].length() + " j is " + j);
           //   if( returnblocks.size() > 20) {
           //    break;
           //    }
           // }

            //returnblocks.addLast(Translate.urlEncodeText(" ","auto","english"));
//it would be nive if you could preserve the "paragraph marker"...
        }
      } catch (Exception ex) {
        ex.printStackTrace();
      }
      return returnblocks;
   }

   public static String averageLangDetectold(String text, int numToTry) {
      String rval = "error";
      try {
        String detectedCode1 = Translate.detect(text.substring(0,200));
        LinkedList<URL> toCheck = new LinkedList();
        while (toCheck.size() < numToTry) {
           int i = (text.length() / numToTry) * toCheck.size();
           int j = text.length();
           //       Sout("\n size is " + text.length() + " on substring " + i + " to "
+ j);
           URL potentialBlock =
Translate.urlEncodeText(text.substring(i,j),"auto","english");
           while ( potentialBlock.toString().length() > TRANSLATOR_LIMIT ) {
              j = j - 20;  //very crude, assumes translator can handle 20 and make
break up words, sentences, etc
              potentialBlock =
Translate.urlEncodeText(text.substring(i,j),"auto","english");
           }
           toCheck.addLast(potentialBlock);
        }
        Hashtable h = new Hashtable();
        while (toCheck.size() > 0 ) {
           String code = Translate.detect(toCheck.removeLast());

           if(h.containsKey(code)) {
              Integer temp = (Integer) h.get(code);
              h.put(code,++temp );
           }
           else {
              h.put(code,new Integer(1));
           }
        }
        if (h.isEmpty()){
           rval = "error while determining lang";
        }
        else if(h.size() == 1) {
           Enumeration e = h.keys();
           rval = (String)e.nextElement();
           Serr(" " + rval + "occured " +  h.get(rval) + " times ");
        }
        else if(h.size() > 1) {
           Serr("multiple languages found! Watch out!");
           Enumeration e = h.keys();
           int max = 0;
           while (e.hasMoreElements()) {
              String curl = (String) e.nextElement();
              Integer curv = (Integer)  h.get(curl);
              if ( max < curv) {
                max = curv;
                rval = curl;
              }
           }
        }

      } catch (Exception ex) {
        ex.printStackTrace();
      }
      return rval;

   }
   public static String averageLangDetect(String text, int numToTry) {
      String rval = "error";
      try {
        String detectedCode1 = Translate.detect(text.substring(0,200));
        LinkedList<URL> toCheck = new LinkedList();
        while (toCheck.size() < numToTry) {
           int i = (text.length() / numToTry) * toCheck.size();
           int j = text.length();
           //      Sout("\n size is " + text.length() + " on substring " + i + " to "
+ j);
           URL potentialBlock =
Translate.urlEncodeText(text.substring(i,j),"auto","english",Translate.DETECT_URL);
           while ( potentialBlock.toString().length() > TRANSLATOR_LIMIT ) {
              j = j - 20;  //very crude, assumes translator can handle 20 and make
break up words, sentences, etc
              potentialBlock =
Translate.urlEncodeText(text.substring(i,j),"auto","english",Translate.DETECT_URL);
           }
           toCheck.addLast(potentialBlock);
        }
        Hashtable h = new Hashtable();
        Hashtable c = new Hashtable();
        while (toCheck.size() > 0 ) {
           JSONObject j = Translate.detectall(toCheck.removeLast());
           //google translate language detect v1.0 supports three responseData strings
           String code = ((JSONObject)j.get("responseData")).getString("language");
           String isrel= ((JSONObject)j.get("responseData")).getString("isReliable");
           String conf = ((JSONObject)j.get("responseData")).getString("confidence");

           //Sout(" " + code + " " + isrel + " " + conf);

           if(h.containsKey(code)) {
              Integer temp = (Integer) h.get(code);
              h.put(code,++temp );
              c.put(code,(Double)c.get(code) + new Double(conf)); //sum the confidence
values
           }
           else {
              h.put(code,new Integer(1));
              c.put(code,new Double(conf));
           }
        }
        if (h.isEmpty()){
           rval = "error while determining lang";
        }
        else if(h.size() == 1) {
           Enumeration e = h.keys();
           rval = (String)e.nextElement();
           Serr(" " + rval + "occured " +  h.get(rval) + " times ");
        }
        else if(h.size() > 1) {
           Serr("multiple languages found! Watch out!");
           Enumeration e = h.keys();
           Double max = 0.0;
           while (e.hasMoreElements()) {
              String curl = (String) e.nextElement();
              Integer curv = (Integer)  h.get(curl);
              Double avec = (Double) c.get(curl) / curv;
              if ( max < avec) {
                max = avec;
                rval = curl;
              }
           }
        }

      } catch (Exception ex) {
        ex.printStackTrace();
      }
      return rval;

   }
   public static void Serr(String s){
           System.err.println(s);
   }

   public static void Serr(String s, int v) {
       if (v < VERBOSITY){
           Serr(s);
        }
   }
   public static void Sout(String s){
           System.out.println(s);
   }

   public static void Sout(String s, int v) {
       if (v < VERBOSITY){
           Sout(s);
        }
   }

   public static String editTimeFormate(String t) {
       Long time = new Long(t);
       String ret = "";
       if( time < 60) {
              time = time;
              ret = "" + time + " seconds";
       }
       else if( time < (60 * 60)) {
              time = time / 60;
              ret = "" + time + " minutes";
       }
       else if( time < (24 * 60 * 60)) {
              time = time / (60 * 60);
              ret = "" + time + " days";
       }

       return ret;
   }
```

```
   public static void usage() {
      Serr("");
      Serr("");
      Serr("this program translates word documents to english using google
translate");
      Serr("");
      Serr("the program requires 1 or more file names as arguments");
      Serr("");
      Serr("Options:");
      Serr(" -f <lang>  forces translation to a chosen language");
      Serr("            must be followed immediately by an ISO 639 language code.
(like en,it,fr,ru...)");
      Serr(" -v         increases verbosity (may be used multiple times)");
      Serr(" -s         'simple' mode.  Only displays the translated text, no extra
information");
       return;
   }

   public static void main(String args[]) throws Exception {

      String convertTo = "auto";

      if (args.length < 1) {
        usage();
        return;
      }
      /*
      List<String> challengeFiles = Arrays.asList(new String[] {
      //"Document1Tx.doc"});
      "Document1Tx.doc", "Document2Tx.doc", "Document3Tx.doc",
      "Document4Tx.doc", "Document5Tx.doc" });
   //   "test.doc"});
       */

      List<String> challengeFiles = new ArrayList();
      //       challengeFiles.add("Document1Tx.doc");
      for(int i=0; i < args.length; i++) {
        if (args[i].equals("-s")) {
               VERBOSITY=0;
         }
        else if (args[i].equals("-v")) {
               VERBOSITY++;
         }
        else if (args[i].equals("-f")) {
           if (args[i+1].length() == 2) {
              Sout("Language forced to " + args[i+1] );
              convertTo = args[i+1];
               i++;
           }
            else {
               usage();
               return;
            }
         }
        else {
           challengeFiles.add(args[i]);
        }
      }

      if( VERBOSITY > 1){
        Sout("verbosity is " + VERBOSITY);
      }


      LinkedList<URL> blocksToTranslate = new LinkedList();
      Iterator it = challengeFiles.iterator();
      while ( it.hasNext() ) {
        String currentFile =  (String)it.next();

        try {
        // InputStream is = new BufferedInputStream(new FileInputStream(currentFile));
        // WordExtractor wd = new WordExtractor(is);
        HWPFDocument doc = new HWPFDocument(new FileInputStream(currentFile));
        //Sout("Summary is : " + doc.getSummaryInformation().toString());
        WordExtractor wd = new WordExtractor(doc);
        //gets all the text
        String text = wd.getText();
        String detectedCode;
        if (convertTo.equals("auto")) {
           detectedCode = averageLangDetect(text,4);
        }
        else {
           detectedCode = convertTo;
        }
        Locale docLocal= new Locale (detectedCode.substring(0,2));
        //since there are limits imposed on the translation service, breaking up in
logical places is better
        //ideally 'sentence' ending could also be found, but this means understanding
punctuation, etc for all languages
        String paragraphs[] = wd.getParagraphText();
        //getFooterText() (and Header) gets a string of footer data
        //getTextFromPieces() is like getText but gets from word file pieces which may
have "extra junk"
        //stripFields() might be able to clean up the text from Pieces

        //    the getMetadataTextExtractor doesn't seem to be inherited as documnets
(neither is getSummaryInfo..)
        //    WordExtractor meta = wd.getMetadataTextExtractor();
        //    Sout("metadata: " + meta.getText());
        //    blocksToTranslate = getChunksDumb(paragraphs);
        blocksToTranslate = getChunksBySentence(paragraphs, docLocal);
           Sout("");
           Sout("File             : " + currentFile, V_BASIC);

           Sout("Revision         : " + doc.getSummaryInformation().getRevNumber(),
V_META);
           Sout("Author           : " + doc.getSummaryInformation().getAuthor(),
V_META);
           Sout("Create Date/Time : " +
doc.getSummaryInformation().getCreateDateTime(), V_META);
           Sout("Keywords         : " + doc.getSummaryInformation().getKeywords(),
V_META);
           Sout("Last Author      : " + doc.getSummaryInformation().getLastAuthor(),
V_META);
           Sout("Last Printed     : " + doc.getSummaryInformation().getLastPrinted(),
V_META);
           Sout("Last Saved       : " +
doc.getSummaryInformation().getLastSaveDateTime(), V_META);
           Sout("Security         : " + doc.getSummaryInformation().getSecurity(),
V_META);
           Sout("Page Count       : " + doc.getSummaryInformation().getPageCount(),
V_META);
```

```
           //Sout("Edit Time         : " + ( editTimeConvert( new
Long(doc.getSummaryInformation().getEditTime())) / 60), V_META);
           String poiBadEditTime =  "" + doc.getSummaryInformation().getEditTime();
           if(poiBadEditTime.length() > 7) {
              Sout("Edit Time         : " +
editTimeFormate(poiBadEditTime.substring(0,poiBadEditTime.length()-7)), V_META);
           }
           Sout("Word Count        : " + doc.getSummaryInformation().getWordCount(),
V_META);
           Sout("Company           : " +
doc.getDocumentSummaryInformation().getCompany(), V_META);
           Sout("Category          : " +
doc.getDocumentSummaryInformation().getCategory(), V_META);
           Sout("Hidden objs       : " +
doc.getDocumentSummaryInformation().getHiddenCount(), V_META);
           Sout("Manager           : " +
doc.getDocumentSummaryInformation().getManager(), V_META);

           String toTranslate = text;
           //  String detectedCode = Translate.detect(toTranslate.substring(0,200));
//simple detection based on head(file)
           System.out.print("Detected Language: " + Language.code2lang(detectedCode) +
"(" + detectedCode );
           if(detectedCode.length() > 2) {
              detectedCode = detectedCode.substring(0,2);
              System.out.print(" - will translate as " + detectedCode);
           }
           Sout(")");
           String titleOrig = doc.getSummaryInformation().getTitle();
           Sout("title len is " + titleOrig.length(), V_EXTRA);
           if(titleOrig.length() > 0 ) {
              Sout("Title             : " + Translate.translate(titleOrig,
detectedCode, "english"), V_META);
           }
           String commentsOrig = doc.getSummaryInformation().getComments();
           Sout("comments len is " + commentsOrig.length(), V_EXTRA);
           if(commentsOrig.length() > 0 ) {
              Sout("Comments          : " + Translate.translate(commentsOrig,
detectedCode, "english"), V_META);
           }
           //we want to detect language based on the body (larger sample set) then use
that language to translate smaller samples (like title, category, etc)
           while ( blocksToTranslate.size() > 0 ) {
              URL currentBlock = blocksToTranslate.removeFirst();
              String detectedCodeCur = Translate.detect(currentBlock);
              //     JSONObject detected = Translate.detectall(currentBlock);
              //    Sout(detected.toString());
              //     String detectedCodeCur =
((JSONObject)detected.get("responseData")).getString("language");
              if(!detectedCode.equals(detectedCodeCur.substring(0,2))) {  //only
compare the first two...
                 Serr("PARAGRAPHS ARE IN DIFFERENT LANGUAGES?! (" + detectedCode + " !=
" + detectedCodeCur.substring(0,2) + ")", V_INFO);
              }

              //String translatedText = Translate.translate(toTranslate, "auto",
"english");
              String translatedText = Translate.translate(currentBlock);
               if (VERBOSITY > 0 ) {
                Sout("Translated Text  : " + translatedText);
              }
               else {
                  System.out.print("" + translatedText);
               }
               if(translatedText.equals(" ")){  //assume that a whole block that's
just a space is a paragraph marker
                  Sout("");
               }
           }


        } catch (java.io.FileNotFoundException ex) {
           Sout( " file " + currentFile + " not found: " + ex.getMessage());
        } catch (Exception ex) {
           ex.printStackTrace();
        }

      }
   }
}
```

Appendix 5 : 401 MSNXMLParser compilation and execution:

```
1: 4/3/2008 11:53:40 AM  (2008-04-03T18:53:40.109Z)
   Original Session id: 1 Original File:udorntani1744684863.xml
   blane -> Zeus
   what are you talking about
 2: 4/3/2008 11:53:52 AM  (2008-04-03T18:53:52.500Z)
   Original Session id: 1 Original File:udorntani1744684863.xml
   Zeus -> blane
   just what i said
 3: 4/3/2008 11:54:22 AM  (2008-04-03T18:54:22.296Z)
   Original Session id: 1 Original File:udorntani1744684863.xml
   blane -> Zeus
   come on bob, quit horsing around, i know it's you.
 4: 4/3/2008 11:54:29 AM  (2008-04-03T18:54:29.140Z)
   Original Session id: 1 Original File:udorntani1744684863.xml
   Zeus -> blane
   ok, got me
 5: 4/3/2008 11:54:39 AM  (2008-04-03T18:54:39.750Z)
   Original Session id: 1 Original File:udorntani1744684863.xml
   Zeus -> blane
   just wanted to see if you'd bite
 6: 4/3/2008 11:55:07 AM  (2008-04-03T18:55:07.515Z)
   Original Session id: 1 Original File:udorntani1744684863.xml
   Zeus -> blane
   i switching back to normal sign on now
 7: 4/3/2008 10:26:24 AM  (2008-04-03T17:26:24.531Z)
   Original Session id: 1 Original File:yogibear19322121292544.xml
   bob -> yogibear1953@hotmail.com
   back yet
 8: 4/3/2008 10:26:35 AM  (2008-04-03T17:26:35.140Z)
   Original Session id: 1 Original File:yogibear19322121292544.xml
   yogibear1953@hotmail.com -> bob
   not yet
 9: 4/3/2008 10:28:02 AM  (2008-04-03T17:28:02.781Z)
   Original Session id: 1 Original File:yogibear19322121292544.xml
   bob -> yogibear1953@hotmail.com
   have to go for a minute
 10: 4/3/2008 10:28:16 AM  (2008-04-03T17:28:16.796Z)
   Original Session id: 1 Original File:yogibear19322121292544.xml
   yogibear1953@hotmail.com -> bob
   ok
 11: 4/3/2008 10:52:01 AM  (2008-04-03T17:52:01.359Z)
   Original Session id: 2 Original File:yogibear19322121292544.xml
   bob -> blane (a DisplayName changed in the transmission likely yogibear1953@hotmail.com to
blane)
   BACK YET?????
 12: 4/3/2008 10:52:28 AM  (2008-04-03T17:52:28.968Z)
```

```
  Original Session id: 2 Original File:yogibear19322121292544.xml
  blane -> bob
  no hold on will you
13: 4/3/2008 11:18:21 AM  (2008-04-03T18:18:21.609Z)
  Original Session id: 2 Original File:yogibear19322121292544.xml
  bob -> blane
  you back yet
14: 4/3/2008 11:59:06 AM  (2008-04-03T18:59:06.218Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  hey im back you ther?
15: 4/3/2008 11:59:13 AM  (2008-04-03T18:59:13.921Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  hey im back man
16: 4/3/2008 11:59:16 AM  (2008-04-03T18:59:16.937Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  hey you on
17: 4/3/2008 11:59:28 AM  (2008-04-03T18:59:28.781Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  yea i was on the can man
18: 4/3/2008 11:59:31 AM  (2008-04-03T18:59:31.546Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  ok
19: 4/3/2008 11:59:44 AM  (2008-04-03T18:59:44.281Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  everything good on your enc?
20: 4/3/2008 11:59:58 AM  (2008-04-03T18:59:58.812Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  yea its ready im ready you?
21: 4/3/2008 12:00:16 PM  (2008-04-03T19:00:16.968Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  Good, I tested out my specialo black powder cake and man oh man
22: 4/3/2008 12:00:36 PM  (2008-04-03T19:00:36.406Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  You didn't blow it all did you?
23: 4/3/2008 12:01:03 PM  (2008-04-03T19:01:03.796Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  I aint that stupid Blaine
```

```
 24: 4/3/2008 12:01:17 PM  (2008-04-03T19:01:17.890Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   blane -> bob
   Hey, you said no names
 25: 4/3/2008 12:02:12 PM  (2008-04-03T19:02:12.546Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   bob -> blane
   Sorry, were even then.  Listen I took a handful of the stuff, put in in that metal pipe with the ball
berrings glued all over and went down to the dump at night.
 26: 4/3/2008 12:03:10 PM  (2008-04-03T19:03:10.453Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   bob -> blane
   Lit that bad mojo off and ran over the hill and dropped 'WHAM'.  Took a quick look and what a hole
and everything standing was shredded.  What a ruswh.
 27: 4/3/2008 12:03:34 PM  (2008-04-03T19:03:34.265Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   blane -> bob
   So that's what thyre talkin about on the news this morning
 28: 4/3/2008 12:03:51 PM  (2008-04-03T19:03:51.593Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   bob -> blane
   News, what news? What u talking about?
 29: 4/3/2008 12:04:45 PM  (2008-04-03T19:04:45.734Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   blane -> bob
   It was all over the news, some kind of explosion at the dump was reported.  They're checking to see if
it as like natural gass or something or some junk somebody threw away
 30: 4/3/2008 12:04:52 PM  (2008-04-03T19:04:52.328Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   bob -> blane
   nuts
 31: 4/3/2008 12:05:29 PM  (2008-04-03T19:05:29.109Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   blane -> bob
   theyre going to figure this out man, that was a astupid play now they got the evidence
 32: 4/3/2008 12:05:52 PM  (2008-04-03T19:05:52.437Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   bob -> blane
   they don't have jack, all they got is a hole and some busted stuff
 33: 4/3/2008 12:06:46 PM  (2008-04-03T19:06:46.156Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   blane -> bob
   no, that was stupid.  They got all this stuff to tell themn what is was and who made it.  I watch those
shows on tv about them CSI dudes and they always figure it out
 34: 4/3/2008 12:07:50 PM  (2008-04-03T19:07:50.359Z)
   Original Session id: 3 Original File:yogibear19322121292544.xml
   blane -> bob
```

```
  whyd u have to do it so close to the city man, why not an out of state test
 35: 4/3/2008 12:24:54 PM  (2008-04-03T19:24:54.546Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  cause mom wouldn't let me have the car last noght and i couldn't drive it out of state even if I had it, no money for gas.  So knock off htat stupid stuff I did what I could.  Least I was smart enought and I tested it out and those tv shors are just that and a bunch of stuff too.
 36: 4/3/2008 12:26:23 PM  (2008-04-03T19:26:23.843Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  Man, I seen what they can do, theyre gonna find us and grill us till we give up the whole thing
 37: 4/3/2008 12:26:29 PM  (2008-04-03T19:26:29.437Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  That'
 38: 4/3/2008 12:27:57 PM  (2008-04-03T19:27:57.234Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  That's garbage so knock it off, well pull this, nobody finds us and were rich.  Aint you tired of being poor, working for sucker money.  Only way they get on to us is if you open your mouth and blab it all over
 39: 4/3/2008 12:28:48 PM  (2008-04-03T19:28:48.953Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  I don't rat, ut maybe we better not.  Least not now.
 40: 4/3/2008 12:30:15 PM  (2008-04-03T19:30:15.968Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  It goes as planned and youre gonna be there too.  I didn't spend all this time and effort for you to chicken out at the last nimute.  If youre too yellow to work this with me ill get somebody else to pull it with.  You just give me the guns and gear u got]
 41: 4/3/2008 12:30:55 PM  (2008-04-03T19:30:55.125Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  I aint no mor yellow thatn you.  You think your so bad, just remember I can wip your tail anyday, did a year ago
 42: 4/3/2008 12:31:25 PM  (2008-04-03T19:31:25.671Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  Then your'e in?
 43: 4/3/2008 12:31:48 PM  (2008-04-03T19:31:48.250Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  You better believe it don't ever call me yellow again
 44: 4/3/2008 12:46:31 PM  (2008-04-03T19:46:31.093Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  Ok, need you like we planned tomorrow night outside the wharehouse.  We go in while the guards are
all at lunch, find the crates of ipods and get them out the door.  Then we set the charges and get the
stuff in the car and get out before the boom
 45: 4/3/2008 12:46:35 PM  (2008-04-03T19:46:35.234Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  What about the fuards
 46: 4/3/2008 12:46:44 PM  (2008-04-03T19:46:44.421Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  guards
 47: 4/3/2008 12:48:15 PM  (2008-04-03T19:48:15.156Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  If we stick to the schedule theyre still at lunc in the front of the building and with the walls the blast
will never get them but all the evidence will be blasted.  Nothing to point them at us and they'll figure
the stuff we stole was deestroyed and probably wont even look for that stuff, just fugure gas blew or
something
 48: 4/3/2008 12:48:23 PM  (2008-04-03T19:48:23.359Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  Man I hop your right
 49: 4/3/2008 12:49:17 PM  (2008-04-03T19:49:17.968Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  I am, and then we sell all those ipods for maybe 300 or 400 bucks each, couple of hundred of them,
and figure the money with a 50 50 split.  Lotta long green, keep you loaded a long time
 50: 4/3/2008 12:49:23 PM  (2008-04-03T19:49:23.734Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  Sweet
 51: 4/3/2008 12:49:48 PM  (2008-04-03T19:49:48.765Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  ]So get a move on, I'll meet you there an hour before party time
 52: 4/3/2008 12:50:07 PM  (2008-04-03T19:50:07.750Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  blane -> bob
  Done, im outta here
 53: 4/3/2008 12:50:24 PM  (2008-04-03T19:50:24.109Z)
  Original Session id: 3 Original File:yogibear19322121292544.xml
  bob -> blane
  Im gone too
 54: 4/3/2008 11:53:16 AM  (2008-04-03T18:53:16.796Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  Zeus -> blane
  hey man whats up long time no talk
 55: 4/3/2008 11:53:40 AM  (2008-04-03T18:53:40.109Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  blane -> Zeus
  what are you talking about
 56: 4/3/2008 11:53:52 AM  (2008-04-03T18:53:52.500Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  Zeus -> blane
  just what i said
 57: 4/3/2008 11:54:22 AM  (2008-04-03T18:54:22.296Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  blane -> Zeus
  come on bob, quit horsing around, i know it's you.
 58: 4/3/2008 11:54:29 AM  (2008-04-03T18:54:29.140Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  Zeus -> blane
  ok, got me
 59: 4/3/2008 11:54:39 AM  (2008-04-03T18:54:39.750Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  Zeus -> blane
  just wanted to see if you'd bite
 60: 4/3/2008 11:55:07 AM  (2008-04-03T18:55:07.515Z)
  Original Session id: 1 Original File:yogibear19534226628795.xml
  Zeus -> blane
  i switching back to normal sign on now
 61: 4/3/2008 10:52:01 AM  (2008-04-03T17:52:01.359Z)
  Original Session id: 2 Original File:yogibear19534226628795.xml
  bob -> blane (a DisplayName changed in the transmission likely yogibear1953@hotmail.com to
blane)
  BACK YET?????
 62: 4/3/2008 10:52:28 AM  (2008-04-03T17:52:28.968Z)
  Original Session id: 2 Original File:yogibear19534226628795.xml
  blane -> bob
  no hold on will you
 63: 4/3/2008 11:18:21 AM  (2008-04-03T18:18:21.609Z)
  Original Session id: 2 Original File:yogibear19534226628795.xml
  bob -> blane
  you back yet
```

Appendix 6 : MSNXMLParser source code.

```
//Tim Vidas, DC3 challenge 2008
import java.io.File;
import org.w3c.dom.Document;
import org.w3c.dom.*;


import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;
import java.util.*;

public class MSNXMLParser{
   private static int VERBOSITY = 5;
   private static boolean POSTGRES = false;
   private static boolean MYSQL = false;
   private static boolean DODROP = false;
   //this really should use java's sql packages and interface with the sql server (or at least
use non-text datatypes)
   private static String postgresSchema = "-- createuser -U postgres 401\n-- createdb -U 401
msn\n-- psql -U 401 msn\nCREATE TABLE messages (\n mid SERIAL UNIQUE,\n date TEXT,\n time TEXT,
\n DateTime TEXT,\n OrigSessionID TEXT,\n fromuser TEXT,\n touser TEXT,\n Style TEXT,\n theText
TEXT,\n OrigFile TEXT,\n NameChangeDetected TEXT,\n NameChangeInfo TEXT,\n PRIMARY KEY (mid)\n
);\n";
   private static String postgresDropTable = "drop table messages;\n";

   public static void Serr(String s){
      System.err.println(s);
   }

   public static void Serr(String s, int v) {
      if (v < VERBOSITY){
        Serr(s);
      }
   }
   public static void Sout(String s){
      System.out.println(s);
   }

   public static void Sout(String s, int v) {
      if (v < VERBOSITY){
        Sout(s);
      }
   }

   public static void usage() {
      Serr("");
      Serr("");
      Serr("this program parses one or more MSN xml log files");
      Serr("these files are typically found in:");
      Serr("  ..\\My Received Files\\LOCALUSER####\\History\\REMOTEUSER####.xml");
      Serr("  where LOCALUSER is the msn account on the local machine");
      Serr("    and REMOTEUSER us the msn account on the remote machine");
      Serr("    and #### is several (10?) numbers that appear to be unique to the msn account");
      Serr("");
      Serr("the program requires 1 or more file names as arguments");
      Serr("");
      Serr("Options:");
      Serr(" -v        increases verbosity (may be used multiple times)");
      Serr(" -V        increases verbosity to Max");
      Serr(" -s        'simple' mode.  Minimal Display.");
      Serr(" -p               generate postgres output to stdout");
      Serr(" -m               generate mysql output to stdout");
      Serr(" -d               drop the messages table prior to inserts (for postgres or mysql)");
      Serr("             (you will lose any information previously stored in this table)");
      Serr("");
      Serr("Examples:");
      Serr("           java MSNXMLParser -V somefile.xml");
      Serr("           java MSNXMLParser -p -d somefile.xml anotherfile.xml > postgres.sql");
      return;
   }

   public static String createMysqlFile(LinkedList<tMessage> m){
       return "MYSQL support not yet implemented";
   }
   public static String createPostgresFile(LinkedList<tMessage> m){
      String ret = "";
      LinkedList<tMessage> pMessages = (LinkedList<tMessage>)m.clone();
      int totalMessageCount = 0;
      Sout(postgresSchema,15);
      String inserts = createInserts(pMessages);

      if(DODROP){
        ret = ret + postgresDropTable;
      }
      ret = ret + postgresSchema + inserts;
      return ret;

   }

   public static String createInserts(LinkedList<tMessage> mlist){
      String ret = "";
      while(mlist.size() > 0) {
        //Sout("X " + totalMessageCount++ + ": " + (mlist.removeFirst()).toString(),15);
        String currInsert = "";
        tMessage am = mlist.removeFirst();
        currInsert = "INSERT INTO messages
(date,time,DateTime,OrigSessionID,fromuser,touser,Style,theText,OrigFile,NameChangeDetected,NameC
hangeInfo) values('";
        currInsert = currInsert + am.Date.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.Time.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.DateTime.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.OrigSessionID.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.From.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.To.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.Style.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.Text.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.OrigFile.replaceAll("'","\\\"");
        currInsert = currInsert + "', '";
        if(am.NameChangeDetected){
          currInsert = currInsert + "true";
        }
        else{
          currInsert = currInsert + "false";
        }
        currInsert = currInsert + "', '";
        currInsert = currInsert + am.NameChangeInfo.replaceAll("'","\\\"");
        currInsert = currInsert + "');";
        Sout("SQL: " + currInsert,15);
        ret = ret + currInsert + "\n ";
      }
      return ret;
   }


   public static void main (String args []){
      if (args.length < 1) {
         usage();
         return;
      }

      try {

         List<String> challengeFiles = new ArrayList();
         for(int i=0; i < args.length; i++) {
            if (args[i].equals("-s")) {
               VERBOSITY=0;
            }
            else if (args[i].equals("-v")) {
               VERBOSITY++;
            }
            else if (args[i].equals("-p")) {
               POSTGRES=true;
            }
            else if (args[i].equals("-m")) {
               MYSQL=true;
            }
            else if (args[i].equals("-d")) {
               DODROP=true;
            }
            else if (args[i].equals("-V")) {
               VERBOSITY=100;
            }
            else {
               challengeFiles.add(args[i]);
            }
         }

         if( VERBOSITY > 1){
            Sout("verbosity is " + VERBOSITY, 15);
         }



         LinkedList<tMessage> theMessages = new LinkedList();
         Iterator it = challengeFiles.iterator();
         while ( it.hasNext() ) {
            String currentFile =  (String)it.next();

               //for each file keep track of to,from pairs in order to detect display name
changes
            List<String> displayNamePairs = new ArrayList();
            boolean haveProcessedOne = false;  //to aide in DisplayNameChange detection


            DocumentBuilderFactory docBuilderFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docBuilderFactory.newDocumentBuilder();
            //Document doc = docBuilder.parse (new File("udorntani.xml"));
            //Document doc = docBuilder.parse (new File("yogi.xml"));
            Document doc = docBuilder.parse (new File(currentFile));

            doc.getDocumentElement ().normalize ();
            Element root = doc.getDocumentElement();
            Sout( "==============================================", 15);
            Sout ("Processing File: " + currentFile, 15);
            Sout ("Root element is " + root.getNodeName(), 15);
            String FirstSession = root.getAttribute("FirstSessionID");
            String LastSession = root.getAttribute("LastSessionID");
            Sout("First session: " + FirstSession + " Last Session: " + LastSession, 15);

            NodeList listOfMessages= doc.getElementsByTagName("Message");
            int totalMessages = listOfMessages.getLength();
            Sout("Total no of messages: " + totalMessages, 15);

            for(int s=0; s<listOfMessages.getLength() ; s++){
               Sout("", 15);
```

```
              Node currMessageNode = listOfMessages.item(s);
              if(currMessageNode.getNodeType() == Node.ELEMENT_NODE){

                 Element firstMessageElement = (Element)currMessageNode;
                 String Date = firstMessageElement.getAttribute("Date");
                 String Time = firstMessageElement.getAttribute("Time");
                 String DateTime = firstMessageElement.getAttribute("DateTime");
                 String SessionID = firstMessageElement.getAttribute("SessionID");
                 Sout("Local timestamp: " + Date + " " + Time + " (" + DateTime + ")", 15);
                 Sout("Local SessionID: " + SessionID, 15);

                 NodeList fromNameList = firstMessageElement.getElementsByTagName("From");
                 Element fromNameElement = (Element)fromNameList.item(0);
                 NodeList fuserNameList = fromNameElement.getElementsByTagName("User");
                 Element fuserNameElement = (Element)fuserNameList.item(0);
                 String fuser = fuserNameElement.getAttribute("FriendlyName");
                 //Sout("From: " + fuser, 15);

                 NodeList toNameList = firstMessageElement.getElementsByTagName("To");
                 Element toNameElement = (Element)toNameList.item(0);
                 NodeList tuserNameList = toNameElement.getElementsByTagName("User");
                 Element tuserNameElement = (Element)tuserNameList.item(0);
                 String tuser = tuserNameElement.getAttribute("FriendlyName");
                 Sout("From: " + fuser + " -> To: " + tuser, 15);


                 NodeList textNameList = firstMessageElement.getElementsByTagName("Text");
                 Element textNameElement = (Element)textNameList.item(0);
                 //NodeList styleNameList = textNameElement.getElementsByTagName("Style");
                 //Element styleNameElement = (Element)styleNameList.item(0);
                 String style = textNameElement.getAttribute("Style");
                 Sout("style: " + style, 15);
                 NodeList TextText = textNameElement.getChildNodes();
                 String text = ((Node)TextText.item(0)).getNodeValue().trim();
                 Sout("Text : " + text, 15);

                 //tMessage amessage = new tMessage();
                 //amessage.settMessage(Date, Time, DateTime, SessionID, fuser, tuser, style,
text);
                 tMessage amessage = new tMessage(Date, Time, DateTime, SessionID, fuser, tuser,
style, text, currentFile);

                 if(!displayNamePairs.contains(fuser + "->" + tuser)){
                        displayNamePairs.add(fuser + "->" + tuser);
                        displayNamePairs.add(tuser + "->" + fuser);  //generic assuptions that
comms are bidirectional

                        //if(theMessages.size() > 1) {  //this needs to be messages per file,
not just messager (to handle multiple files
                        if(haveProcessedOne) {  //this needs to be messages per file, not just
messager (to handle multiple files
                              Serr("-- DisplayName change detected. " + fuser + "->" + tuser + "
has not been seen before");

                              amessage.NameChangeDetected = true;

                              //at this point we could implement and edit distance or Levenshtein
distance check on from->to
                              //but we just run through the message list looking for a half match
                               Iterator mit = theMessages.iterator();
                               while ( mit.hasNext() ) {
                                  tMessage m =  (tMessage) mit.next();
                                  //if display name matches but the other doesn't, then we have a
likely canidate for a name change
                                  if( fuser.equals(m.From)){
                                         if(! tuser.equals(m.To)){
                                             Serr("Likely display name change: " + m.To + " is
now " + tuser);
```

```
                                             amessage.NameChangeInfo = "" + m.To + " to " +
tuser;
                                             break;
                                     }
                                  }
                                  if( fuser.equals(m.To)){
                                     if(! tuser.equals(m.From)){
                                             Serr("Likely display name change: " + m.From + " is
now " + tuser);
                                             amessage.NameChangeInfo = "" + m.From + " to " +
tuser;
                                             break;
                                     }
                                  }
                              }
                      }
                      haveProcessedOne = true;
                 }
                 theMessages.addLast(amessage);
              }
           }

        }

        //at this point all the messages from all the files are stored in theMessages list.  We
can print, or whatever

        if (MYSQL) {
           //create mysql output
           Sout(createMysqlFile(theMessages));
        }
        else if (POSTGRES) {
           //create postgres output
           Sout(createPostgresFile(theMessages));
        }
        else {

           //print them with asc ID
           int totalMessageCount = 0;
           while(theMessages.size() > 0) {
              Sout(" " + totalMessageCount++ + ": " +
(theMessages.removeFirst()).toFormattedString());
           }
        }

      }catch (java.io.FileNotFoundException fnf) {
        System.err.println(" bad filename given " + fnf.getMessage());

      }catch (SAXParseException err) {
        System.err.println ("SAX Parse error" + " " + err.getLineNumber () + ", uri " +
err.getSystemId ());
        System.err.println(" " + err.getMessage ());

      }catch (SAXException e) {
        Exception x = e.getException ();
        e.printStackTrace ();

      }catch (Throwable t) {
        t.printStackTrace ();
      }

   }



}
```